Dissertation on

"STOCK PRICE PREDICTION USING LSTM"

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology
in
Computer Science & Engineering

UE19CS390B – Capstone Project Phase - 2

Submitted by:

DISHA BS	**PES2UG19CS116**
J THANISH VISHAAL	**PES2UG19CS155**
PRATHEEK	**PES2UG19CS291**
S KALYAN	**PES2UG19CS341**

Under the guidance of

Prof. Shilpa S
Assistant Professor
PES University

June - Nov 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING
PES UNIVERSITY
(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

FACULTY OF ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

'STOCK PRICE PREDICTION USING LSTM'

is a bonafide work carried out by

DISHA BS	**PES2UG19CS116**
J THANISH VISHAAL	**PES2UG19CS155**
PRATHEEK	**PES2UG19CS291**
S KALYAN	**PES2UG19CS341**

In partial fulfilment for the completion of seventh semester Capstone Project Phase - 2 (UE19CS390B) in the Program of Study -Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period June 2022 – Nov. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The dissertation has been approved as it satisfies the 7th semester academic requirements in respect of project work.

Signature	Signature	Signature
Prof.Shilpa S	Dr. Sandesh B J	Dr. B K Keshavan
Assistant Professor	Chairperson	Dean of Faculty

External Viva

Name of the Examiners	**Signature with Date**
1. ____________________________	____________________________
2. ____________________________	____________________________

DECLARATION

We hereby declare that the Capstone Project Phase - 2 entitled **stock price prediction using LSTM** has been carried out by us under the guidance of Prof. Shilpa S, Assistant Professor and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester June – Nov. 2022. The matter embodied in this report has not been submitted to any other university or institution for the award of any degree.

PES2UG19CS116	**DISHA BS**
PES2UG19CS155	**J THANISH VISHAAL**
PES2UG19CS291	**PRATHEEK**
PES2UG19CS341	**S KALYAN**

ACKNOWLEDGEMENT

I would like to express my gratitude to Prof. Shilpa S, Assistant Professor Department of Computer Science and Engineering, PES University, for her continuous guidance, assistance, and encouragement throughout the development of this UE19CS390B -Capstone Project Phase – 2.

I am grateful to the Capstone Project Coordinator, Dr. Sarasvathi V, Professor and Dr. Sudeepa Roy Dey, Associate Professor, for organizing, managing, and helping with the entire process.

I take this opportunity to thank Dr. Sandesh B J, Chairperson, Department of Computer Science and Engineering, PES University, for all the knowledge and support I have received from the department. I would like to thank Dr. B.K. Keshavan,Dean of Faculty, PES University for his help.

I am deeply grateful to Dr. M. R. Doreswamy, Chancellor, PES University, Prof. Jawahar Doreswamy, Pro Chancellor – PES University, Dr. Suryaprasad J, Vice-Chancellor, PES University for providing to me various opportunities and enlightenment every step of the way. Finally, this project could not have been completed without the continual support and encouragement I have received from my family and friends.

ABSTRACT

In this project we attempt to implement machine learning approach to predict stock prices. Machine learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, machine learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades. There are two types of stocks. Intraday trading by the commonly used term "day trading". Interday traders hold securities positions from at least one day to the next and often for several days to weeks or months.

LSTMs are very powerful in sequence prediction problems because they’re able to store past information. This is important in our case because the previous price of a stock is crucial in predicting its future price. While predicting the actual price of a stock is an uphill climb, we can builda model that will predict whether the price will go up or down.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Throughout the preceding two years. owing in large part to technological developments, stock trading has attracted a lot of attention. Investors look for strategies and tactics that could increase profits while reducing risk. SMP is potentially a type of statistical forecasting that analyses past dataand forecasts future values.

A meeting of stock buyers and sellers could be referred to as a stock exchange, sometimes known as a share market. Equity crowd funding platforms are a market where investors can purchase shares of individual businesses. Additionally, to other asset classes, common equality shares are also listed on stock exchanges. For instance, corporate bonds and convertible bonds.

Many time series forecasting algorithms have proven their effectiveness in practice. Time series forecasting is a common technique widely used in many real-world applications such as weather forecasting and financial market forecasting. The most popular algorithms today are based on recurrent neural networks (RNNs) and their special types long short-term memory (LSTM) and gated recurrent units (GRU). Time series forecasting is a common technique widely used in many real-world applications such as weather forecasting and financial market forecasting. The stock market is a representative field that presents time-series data, and many researchers have studied it and proposed various models. Using continuous data over a period of time to predict the next unit time result. This project uses an LSTM model to predict stock prices.

CHAPTER 2

PROBLEM STATEMENT

The aim of the project is to predict rise and fall of stock prices thorough RNN and LSTM. The project assists users in deciding whether to buy or sell stocks on the stock market. Users can connect in to the web server and view number of graphs before making their decisions. Stock prices are highly volatile and its nature varies depending upon various factors.

CHAPTER-3

LITERATURE SURVEY

3.1 Applications of Neutral Networks

3.1.1 Introduction

Computational intelligence is receiving a lot of interest in industry due to its potential application.one such example of CI is ANN artificial neural network. ANNs can learn and are taught to make intelligent decisions based on data.

3.1.2 Objective and Methodology

Training for GRNN is simple. Training inputs and targets are converted into input weights and output weights. There is equality in hidden neurons and training samples due to the associative memory of GRNN However, if there are a large number of training samples, this training method is inefficient. error-based approach is one such solution. The algorithm will determine whether it is necessary to include the input in the training prior to training GRNN with an input based on prediction error. The inputs will be used to train GRNN only If they prediction error without that input is bigger than a precise threshold,

3.1.3 Advantages

• Simple to implement and useful for real-time prediction.

• Approach to quick training.

• Both linear and non-linear functional regressions have good accuracy.

3.1.4 Limitations

• More RAM is required to store the mode.

Because of its enormous size, it can be computationally expensive

3.2 Workload Forecasting with Long Short-Term Memory

3.2.1 Introduction

Despite its many benefits, cloud computing has a number of drawbacks and difficulties, such as power consumption and dynamic resource scalability. Vulnerabilities and cost come into picture. High accuracy is obtained by decreasing the mean squared error, considering the empirical results. How the system's workload will change in the future is the objective of this prediction. This allows you to answer questions like how many emails will arrive in the near future. Because the past is so important in projecting the future, it is crucial to examine the datacenter's history.

3.2.2 Objective and Methodology

The objective is to locate a pattern in the historical data. future workload is anticipated using this workload pattern. n continuous past samples are used to forecast the load. A continual neural network (RNN) may be a system that loops along many networks. the data is preserved because of the whorled networks. information and input from the previous network are fed into to loop, conducts the required action, and produces output whereas additionally causation the information toensuing network

3.2.3 Advantages

• It has a high level of accuracy since it analyses the data for hidden patterns.

• It retains information for a long period of time.

3.2.4 Limitations

• indexing the memory while writing and retrieving data using memory cells is impossible

3.3 Machine Learning for Stock Price Prediction

3.3.1 Introduction

Machine learning (ML) is now an effective tool for managing investments accurately in the financial markets. To improve performance this model is responsible as investors in handling and managing investment ML allows Securities investments in the financial sector. It would be straightforward to predict costs mistreatment solely a couple of parameters, however the results may be off since some aspects that were missed might also be vital in understanding stock cost. A number of variables, like economic growth, will influence the evaluation of specific stocks. To manually analyze each part is difficult. Therefore, it'd be preferred if there have been resources to help within the examination of this information

3.3.2 Advantages

• High capacity to deal with non-linear patterns that are complex.

• Modeling the relationship with high accuracy in data groups mode. • It is the most effective social science forecasting technique. • Strong adaptability and flexible nonlinear modelling capability • Can accurately model non-linearity, and the results are simple to understand.

3.3. 3 Limitations

• Doesn't work well with nonlinear time series; the model may not be suitable for another series if it was created for the first.

• More information is needed, and it is subject to noise.

• ANNs only provide projected goal values for some unknown variables in the absence of variance information needed to evaluate the forecast.

• The choice of parameters has an impact on over-fitting.

3.4 Estimation and prediction based on recorded values

3.4.1 Introduction

The neural network was trained using information from a real-world cast iron foundry in a previous application. The network was fed a set of fresh casting inputs after training in order for it to calculate the percentage of defectives. The actual trial outcomes were discovered to be remarkably similar to the anticipated outcomes. Back propagation (BP) neural networks have been utilized to tackle practical issues that are too complex for conventional approaches. A generalization of the delta rule, which is used to train nonlinear multi-layer feed forward neural networks, is the BP neural network. These investigations sought to identify the most effective neural network type for prediction.

3.4.2 Advantages

• It primarily focuses on data forecasting based on stored value.

• Non-linear modelling skill that is flexible

• High learning precision and adaptability

• For predicting complicated non-linear systems, fast response and parallel processing capabilities are required.

3.4.3 Limitations

• Noise sensitivity

• Its performance is determined by the original values.

• Converging quickly to a local minimum

• Convergent speed is slow.

3.5 Predictions with neural networks

3.5.1 Introduction

The main intention of study is to use artificial neural network (ANN) to approach and forecast global solar radiation (GSR) on exterior of three Egyptian towns. Performance of ANN models are investigated using statistical metrics. These criteria show that the second algorithm produces superior outcomes to the first. A precise and successful tactic is an ANN-based model.

3.5.2 Objective and Methodology

• ANNs are used in this model together with non-time, time, and financial time series.

• It primarily focuses on forecasting and stock classification.

3.5.3 Advantages

• High capacity to deal with non-linear patterns that are complex.

• modelling accuracy that depends on both of the linear process and non-liner process

• The model is powerful and might handle information that's clamant or missing.

3.5.4 Limitations

• Over fitting

• Sensitive to parameter setting

CHAPTER 4

PROJECT REQUIREMENTS SPECIFICATION

4.1 Hardware

A ram of 4 GB with 500 GB storage, a CPU of 2 GHZ or faster and the architecture of either 32 or 64 bits.

4.2 Software

- Python
- Jupyter notebook
- Any operating system MAC, Linux, windows 7 or above

4.3 Functional Requirment

When developing stock prediction software, functional requirements include:

• The software must accept stock data CSV files from the Yahoo Financial website dataset as input.

• The software should perform preprocessing (such as checking for irregular data sets) on the training dataset model.

• Web-App must use LSTM algorithm as the primary unit of the software.

• Processing the given datasets by giving out the best Stock Closing Price result.

4.4 Non-Functional Requirement

- Ease of use: We define our software interface in terms of making it easy for all kinds of stock traders and other stock market stakeholders to understand the stock forecasting software interface.
- Efficiency: Maintains the highest possible accuracy of stock closing prices in the least minimum time possible using stock datasets.
- Performance: A attribute of stock forecasting software that tells its feedback from different user actions.

CHAPTER 5

SYSTEM DESIGN

5.1 Use Case Diagram

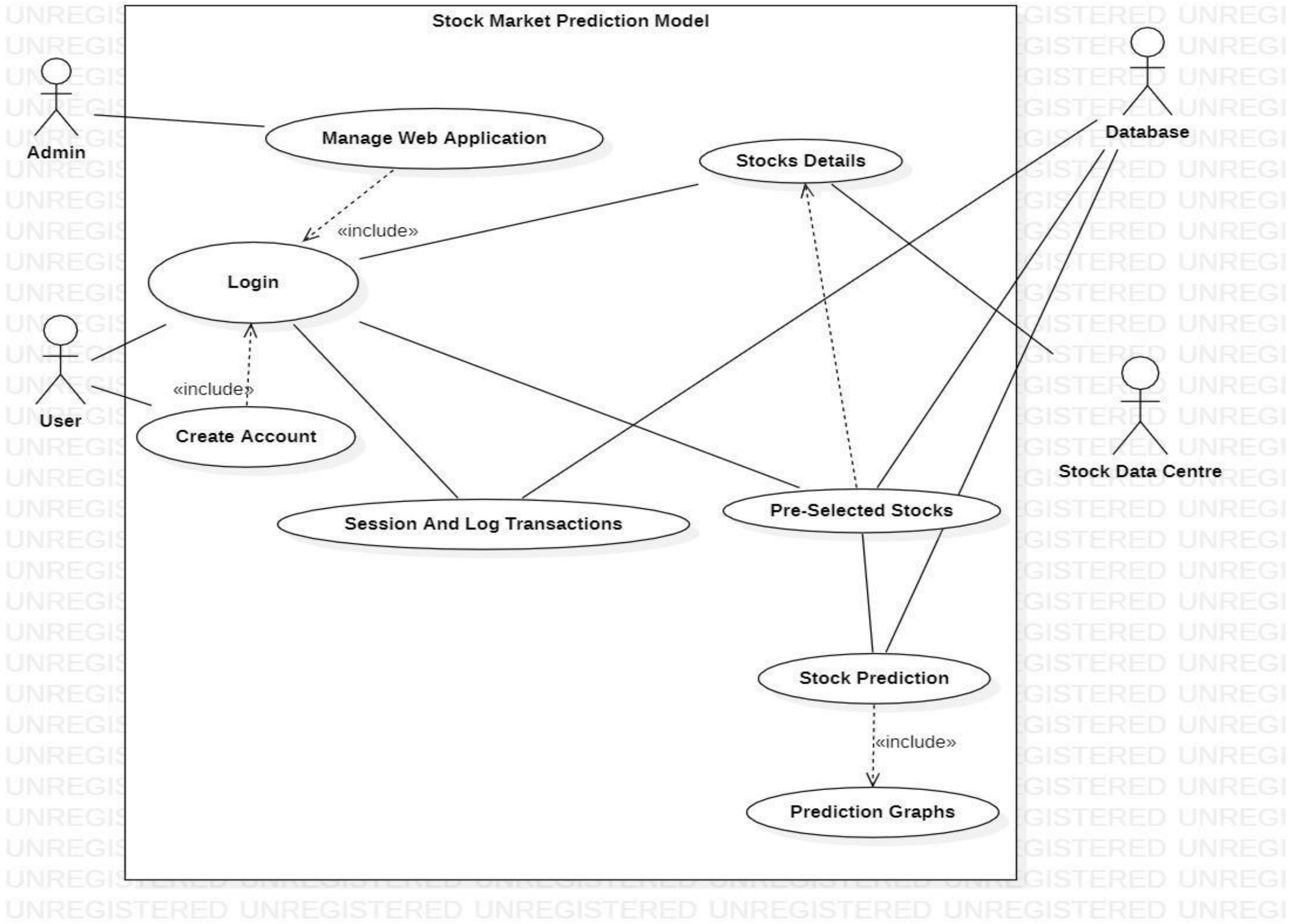


Fig:5.1 Use Case Diagram

5.2 Class Diagram

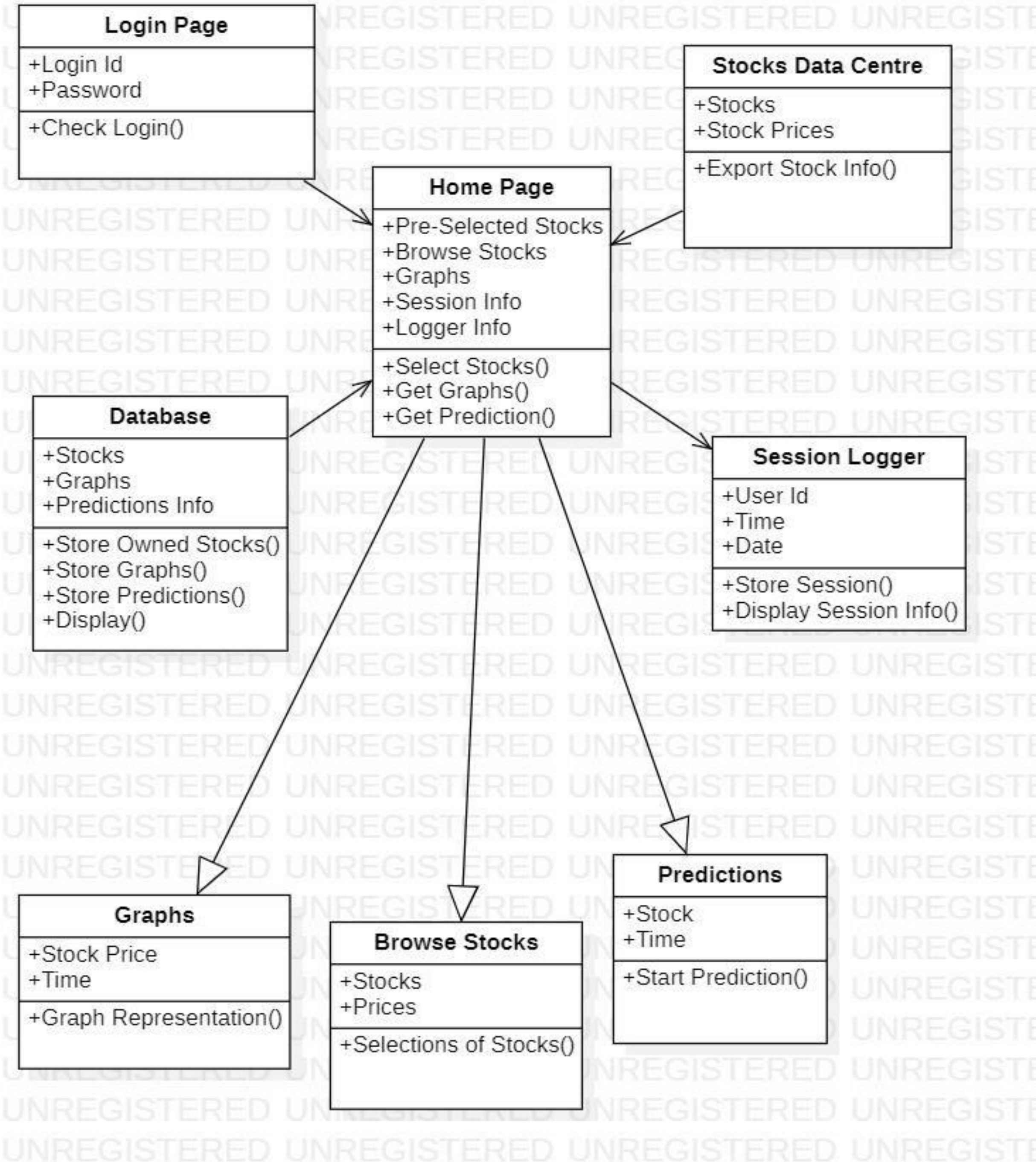


Fig: 5.2 Class Diagram

5.3 Sequence Diagram

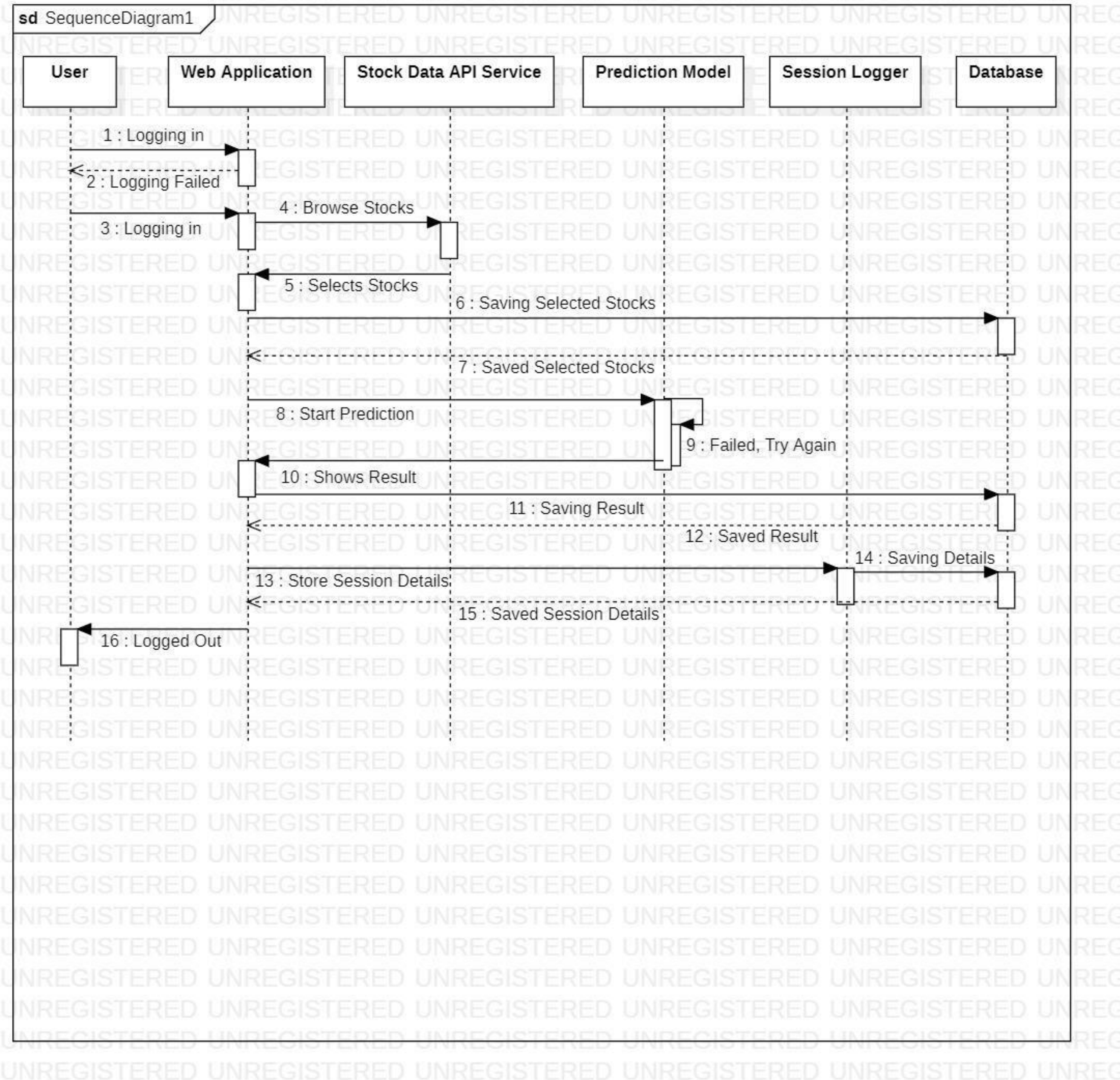


Fig: 5.3 Sequence Diagram

5.4 Deployment Diagram

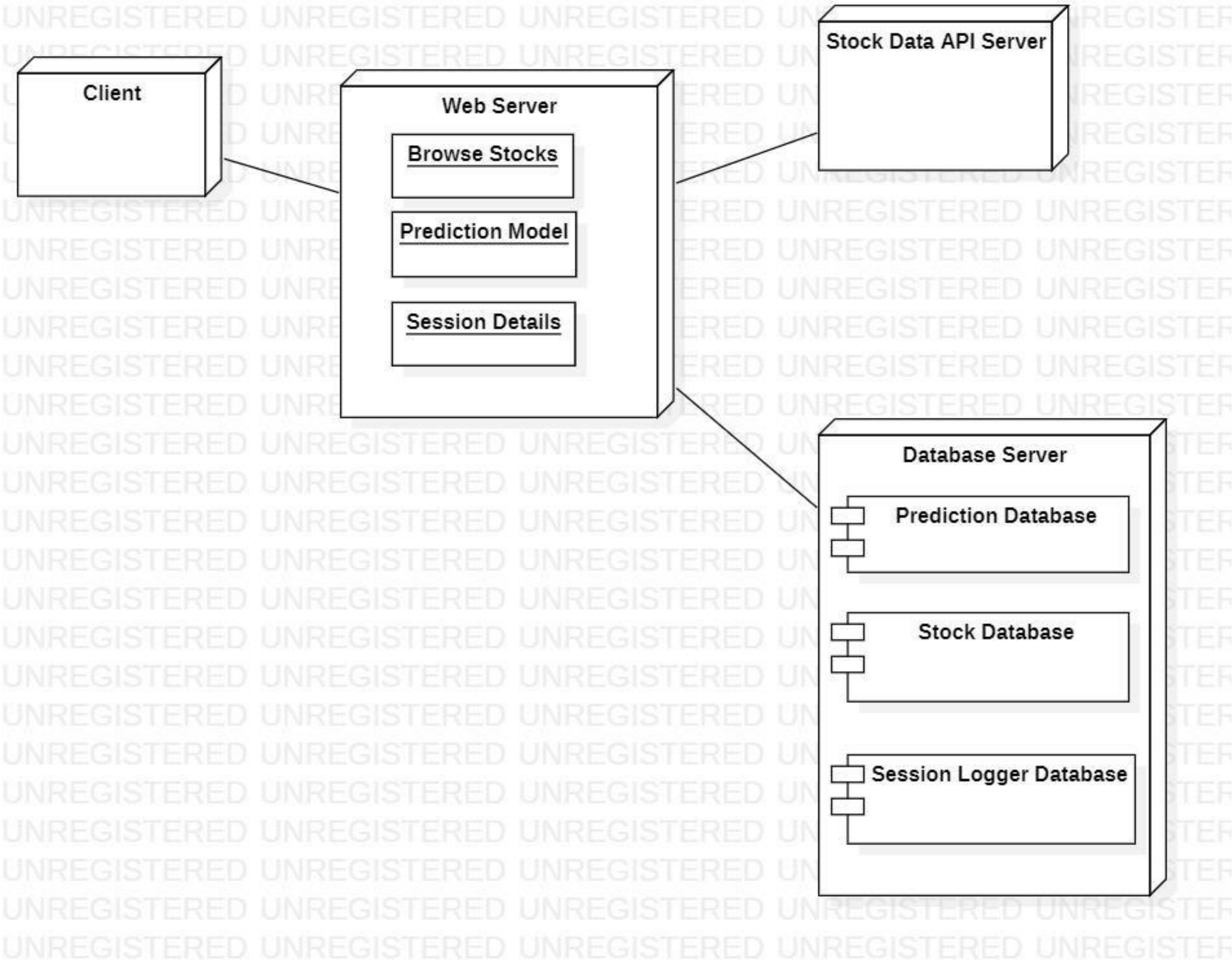


Fig: 5.4 Deployment Diagram

CHAPTER 6

PROPOSED METHODOLOGY

6.1 Basic Approach and Algorithms

- RNN (Recurrent Neural Network)

Recurrent Neural Networks (RNNs) are neural networks that are mostly used for preprocessing datasets such as time series. It is used for stock price prediction. The logic behind this is that after a certain sequence, the price is remembered and the model gains experience based on that pattern. At a high level, the RNN generally uses ‘for’ loop to iterate through time steps of the given datasets sequence while keeping a constant state that stores information about the time steps appeared till now. Since RNNs can only retain sequence patterns for a short time, we switch to LSTMs, which can remember patterns in both short-term and long-term memory.

- LSTM (Long-Term Short-Term Memory)

LSTM is commonly used for machine learning regression and time series forecasting. It distinguishes LSTM neural networks from other neural networks so that data can be stored for long periods of time. The

LSTM is mostly made to avoid longer period dependency errors. Keeping the required information for a longer period of time is effectively its general behaviour.

6.1 Drawbacks of Approach

6.1.1 RNN

- Training RNNs
- Vanishing or Exploding Gradient Issues
- RNNs Cannot Stack
- Slow and Complex Training Procedure
- Longer Sequences Are Difficult to Handle

6.1.2 LSTM

- LSTM takes a longer period of time to train the datasets
- LSTM requires a lot of storage to train the datasets
- LSTM can easily overfit
- Dropout can be much difficult to implement with LSTM
- LSTM is highly volatile to various random weight constants which are initialized throughout the process

6.2 Advantages of Using RNN And LSTM Over Other Models

- The main advantage is time.
- We can use an LSTM in almost no time and start making predictions. Just transform the data from a one-dimensional array to a two-dimensional by lagging the time series for a number of steps and then start your train-test-validation.
- A nice method for having lower forecasting errors with LSTM is using STL decomposition. This way you can decompose the time series in trend, seasonal and residual and by performing forecast on each of those components and summing your results you can make your predictions much more robust.

6.3 Difference Between RNN And LSTM

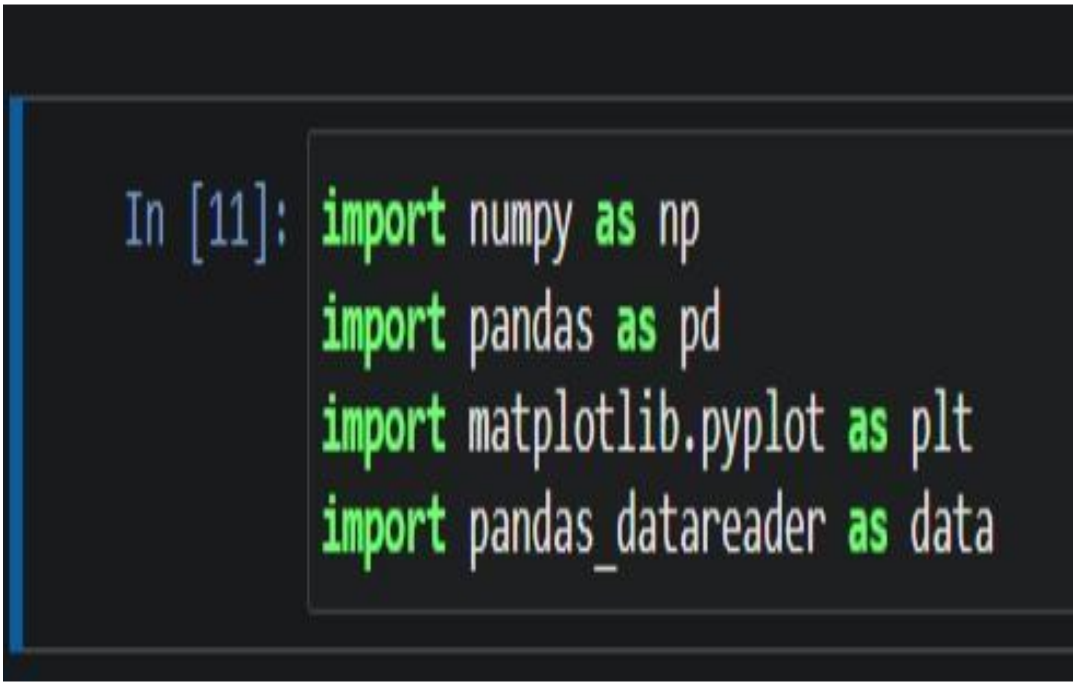
The main difference between RNN and LSTM will be that the information stored in the memory is for a longer period of time. This is the advantage of LSTM over RNN. This is because LSTMs can process information in memory for longer periods of time than RNNs. But the question is, what makes LSTM different from RNN? RNNs give LSTMs the ability to maintain long-term temporal dependencies (memorize information over time).

CHAPTER 7

IMPLEMENTATION AND PSEUDOCODE

7.1 Pseudocode

- **Importing the required libraries**



```
In [11]: import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import pandas_datareader as data
```

Fig:7.1.1 Importing the required libraries

● Reading the data

```
In [12]: start = '2010-01-01'
         end = '2022-12-06'

         df = data.DataReader('AAPL', 'yahoo', start, end)
         df.head()
```

Out[12]:

Date	High	Low	Open	Close	Volume	Adj Close
2009-12-31	7.619643	7.520000	7.611786	7.526071	352410800.0	6.415357
2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.515213
2010-01-05	7.699643	7.616071	7.664286	7.656429	601904800.0	6.526477
2010-01-06	7.686786	7.526786	7.656429	7.534643	552160000.0	6.422666
2010-01-07	7.571429	7.466071	7.562500	7.520714	477131200.0	6.410791

Fig: 7.1.2. Reading the data

- **Resting index**
- **Dropping unwanted**
- **Plotting graph for closing price**

Out[14]:

	Date	High	Low	Open	Close	Volume	Adj Close
0	2009-12-31	7.619643	7.520000	7.611786	7.526071	352410800.0	6.415357
1	2010-01-04	7.660714	7.585000	7.622500	7.643214	493729600.0	6.515213
2	2010-01-05	7.699643	7.616071	7.664286	7.656429	601904800.0	6.526477
3	2010-01-06	7.686786	7.526786	7.656429	7.534643	552160000.0	6.422666
4	2010-01-07	7.571429	7.466071	7.562500	7.520714	477131200.0	6.410791

In [15]:
```
df = df.drop(['Date', 'Adj Close'], axis = 1)
df.head()
```

Out[15]:

	High	Low	Open	Close	Volume
0	7.619643	7.520000	7.611786	7.526071	352410800.0
1	7.660714	7.585000	7.622500	7.643214	493729600.0
2	7.699643	7.616071	7.664286	7.656429	601904800.0
3	7.686786	7.526786	7.656429	7.534643	552160000.0
4	7.571429	7.466071	7.562500	7.520714	477131200.0

In [16]:
```
plt.plot(df.Close)
```

Out[16]: [<matplotlib.lines.Line2D at 0x28c52ac5340>]

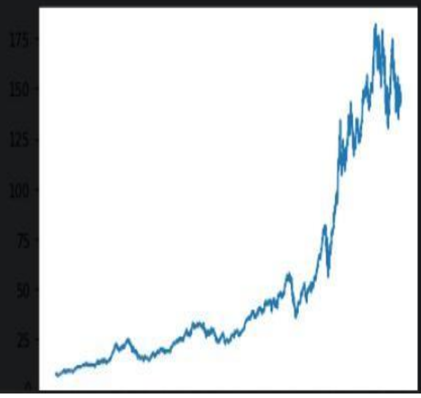


Fig:7.1.3. plotting graph for closing price

- **Moving average of 100 days for given dataset**
- **Plotting graph against moving average and closing price**

```
In [17]: ma100 = df.Close.rolling(100).mean()
         ma100

Out[17]: 0              NaN
         1              NaN
         2              NaN
         3              NaN
         4              NaN
                    ...
         3251    152.911499
         3252    152.935999
         3253    152.959199
         3254    152.940799
         3255    152.868199
         Name: Close, Length: 3256, dtype: float64

In [18]: plt.figure(figsize = (12,6))
         plt.plot(df.Close)
         plt.plot(ma100, 'red')

Out[18]: [<matplotlib.lines.Line2D at 0x28c52afb790>]
```

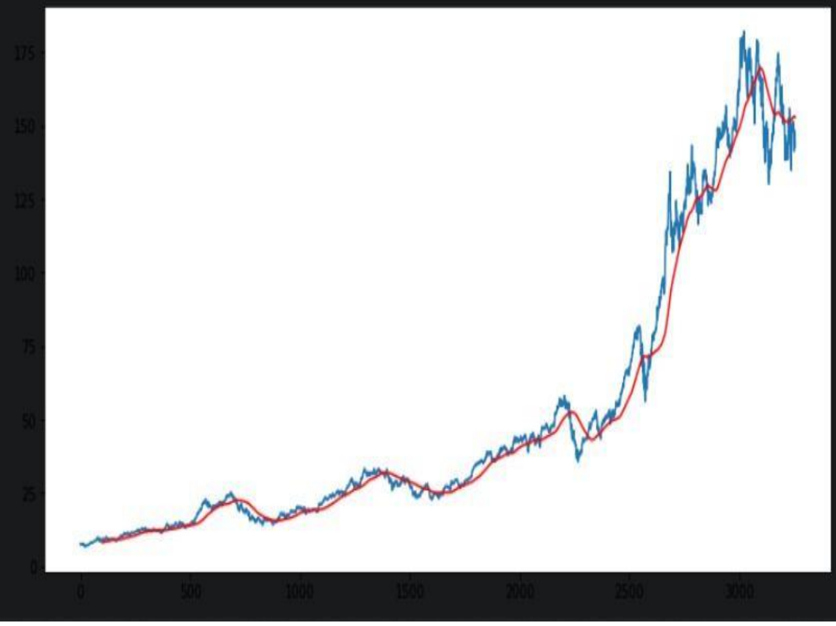


Fig:7.1.4. 100 days moving average

- **Moving average of 200 days for given dataset**
- **Plotting graph against moving average and closing price**

```
In [19]: ma200 = df.Close.rolling(200).mean()
         ma200

Out[19]: 0              NaN
         1              NaN
         2              NaN
         3              NaN
         4              NaN
                   ...
         3251    154.21405
         3252    154.09165
         3253    153.96795
         3254    153.85670
         3255    153.73475
         Name: Close, Length: 3256, dtype: float64

In [20]: plt.figure(figsize = (12,6))
         plt.plot(df.Close)
         plt.plot(ma100, 'red')
         plt.plot(ma200, 'green')

Out[20]: [<matplotlib.lines.Line2D at 0x28c524afa90>]
```

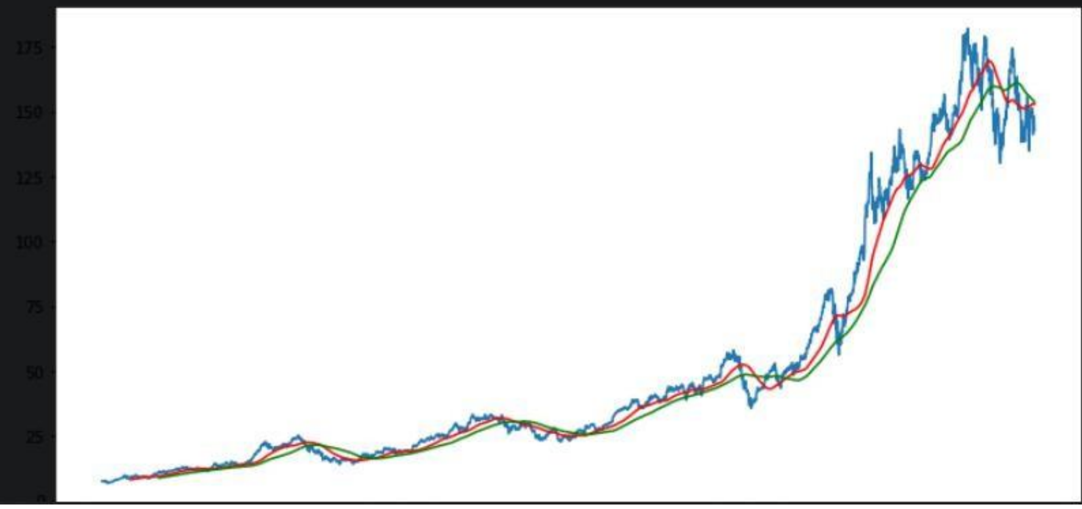


Fig:7.1.5. 200 days moving average

● Splitting data into training and testing

In [22]:
```
# Splitting Data into Training and Testing

data_training = pd.DataFrame(df['Close'][0:int(len(df)*0.70)])
data_testing = pd.DataFrame(df['Close'][int(len(df)*0.70): int(len(df))])

print(data_training.shape)
print(data_testing.shape)
```

```
(2279, 1)
(977, 1)
```

In [23]:
```
data_training.head()
```

Out[23]:

	Close
0	7.526071
1	7.643214
2	7.656429
3	7.534643
4	7.520714

In [24]:
```
data_testing.head()
```

Out[24]:

	Close
2279	38.480000
2280	38.174999
2281	39.439999
2282	39.075001

Fig:7.1.6. training and testing datasets

- **Shaping the training data set**

```
In [25]: from sklearn.preprocessing import MinMaxScaler
         scaler = MinMaxScaler(feature_range=(0,1))

In [26]: data_training_array = scaler.fit_transform(data_training)
         data_training_array

Out[26]: array([[0.01304067],
                [0.01533047],
                [0.01558878],
                ...,
                [0.62757949],
                [0.63227082],
                [0.61506938]])

In [27]: x_train = []
         y_train = []

         for i in range(100, data_training_array.shape[0]):
             x_train.append(data_training_array[i-100: i])
             y_train.append(data_training_array[i, 0])

In [28]: x_train, y_train = np.array(x_train), np.array(y_train)
```

Fig: 7.1.7. Shaping the training data set

- **Creating the stock price prediction model**

```
In [29]: # ML Model

         from keras.layers import Dense, Dropout, LSTM
         from keras.models import Sequential

In [30]: model = Sequential()
         model.add(LSTM(units = 50, activation = 'relu', return_sequences = True,
                        input_shape = (x_train.shape[1], 1)))
         model.add(Dropout(0.2))

         model.add(LSTM(units = 60, activation = 'relu', return_sequences = True))
         model.add(Dropout(0.3))

         model.add(LSTM(units = 80, activation = 'relu', return_sequences = True))
         model.add(Dropout(0.4))

         model.add(LSTM(units = 120, activation = 'relu'))
         model.add(Dropout(0.5))

         model.add(Dense(units = 1))

In [31]: model.summary()

         Model: "sequential"
         _________________________________________________________________
          Layer (type)                Output Shape              Param #
         =================================================================
          lstm (LSTM)                 (None, 100, 50)           10400

          dropout (Dropout)           (None, 100, 50)           0

          lstm_1 (LSTM)               (None, 100, 60)           26640

          dropout_1 (Dropout)         (None, 100, 60)           0

          lstm_2 (LSTM)               (None, 100, 80)           45120

          dropout_2 (Dropout)         (None, 100, 80)           0

          lstm_3 (LSTM)               (None, 120)               96480

          dropout_3 (Dropout)         (None, 120)               0

          dense (Dense)               (None, 1)                 121

         =================================================================
         Total params: 178,761
         Trainable params: 178,761
         Non-trainable params: 0
         _________________________________________________________________
```

Fig:7.1.8. creating the prediction model

- **Compiling the stock price prediction model**

```
In [32]: model.compile(optimizer='adam', loss = 'mean_squared_error')
         model.fit(x_train, y_train, epochs = 50)

Epoch 1/50
69/69 [==============================] - 12s 121ms/step - loss: 0.0270
Epoch 2/50
69/69 [==============================] - 8s 117ms/step - loss: 0.0055
Epoch 3/50
69/69 [==============================] - 8s 110ms/step - loss: 0.0058
Epoch 4/50
69/69 [==============================] - 8s 110ms/step - loss: 0.0048
Epoch 5/50
69/69 [==============================] - 8s 110ms/step - loss: 0.0048
Epoch 6/50
69/69 [==============================] - 8s 116ms/step - loss: 0.0044
Epoch 7/50
69/69 [==============================] - 8s 116ms/step - loss: 0.0045
Epoch 8/50
69/69 [==============================] - 8s 118ms/step - loss: 0.0043
Epoch 9/50
69/69 [==============================] - 8s 115ms/step - loss: 0.0036
Epoch 10/50

In [33]: model.save('keras_model.h5')
```

Fig:7.1.9. compiling the stock data

- **Shaping the testing datasets**

```
In [38]: x_test = []
         y_test = []

         for i in range(100, input_data.shape[0]):
             x_test.append(input_data[i-100: i])
             y_test.append(input_data[i, 0])

In [39]: x_test, y_test = np.array(x_test), np.array(y_test)
         print(x_test.shape)
         print(y_test.shape)

         (977, 100, 1)
         (977,)

In [40]: # Predictions
         y_predicted = model.predict(x_test)

         31/31 [==============================] - 2s 33ms/step

In [41]: y_predicted.shape

Out[41]: (977, 1)

In [42]: scaler.scale_

Out[42]: array([0.00682769])
```

Fig:7.1.10. shaping the testing dataset

● Plotting final graph

```
In [43]: plt.plot(df.Close)

Out[43]: [<matplotlib.lines.Line2D at 0x28c7095b220>]
```


```
In [45]: scaler_factor = 1/0.02099517
         y_predicted = y_predicted * scaler_factor
         y_test = y_test * scaler_factor

In [46]: plt.figure(figsize=(12,6))
         plt.plot(y_test, 'blue', label = 'Original Price')
         plt.plot(y_predicted, 'red', label = 'Predicted Price')
         plt.xlabel('Time')
         plt.ylabel('Price')
         plt.legend()
         plt.show

Out[46]: <function matplotlib.pyplot.show(close=None, block=None)>
```

Fig:7.1.11. Plotting final graph

- **Prediction for next 30 days**

```
x_input = input_data[977:].reshape(1, -1)
temp_input = list(x_input)
temp_input = temp_input[0].tolist()

# demonstrate prediction for next 10 days

lst_output = []
n_steps = 100
i = 0
while (i < 30):

    if (len(temp_input) > 100):
        # print(temp_input)
        x_input = np.array(temp_input[1:])
        print("{} day input {}".format(i, x_input))
        x_input = x_input.reshape(1, -1)
        x_input = x_input.reshape((1, n_steps, 1))
        # print(x_input)
        yhat = model.predict(x_input, verbose=0)
        print("{} day output {}".format(i, yhat))
        temp_input.extend(yhat[0].tolist())
        temp_input = temp_input[1:]
        # print(temp_input)
        lst_output.extend(yhat.tolist())
        i = i+1
    else:
        x_input = x_input.reshape((1, n_steps, 1))
        yhat = model.predict(x_input, verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i = i+1
```

Fig:7.1.12. Prediction for next 30 days

7.2 Implementation

Technologies Used:

- Operating Systems - Microsoft Windows 10/11, MAC and Linux
- Programming Languages - Python
- Frontend Frameworks - Streamlit
- Web Server Software - Python Flask Server
- Machine Learning Libraries - Pandas, Pandas_datareader
- NumPy, SKlearn, Matplotlib, Keras.models

Dataset Collection:

Dataset collection will play out an integral part in any project. For this project, datasets were collected from a 3rd party API i.e., Yahoo Finance. The main advantages of using this API for this project are: -

- Provides robust, powerful and reliable inventory information.
- Provides more than 20 years of data.
- Provides different categories of fundamental and technical stock data.
- Provides JSON/CSV format for easy integration with Python, R, PHP and many other web APIs.

The process of data collection is critical to any project. Therefore, in order to retrieve accurate and valid data, the Streamlit app sends user requests to the Python Flask middleware. The Python Flask API is integrated with third party APIs. B. Yahoo Finance integrated to collect required data. This data is in CSV file format. The Python Flask API returns data to the Streamlit app to fulfill user requests. During this process, the data was checked, transformed, corrected and prepared for various financial visualizations

CHAPTER 8

RESULTS AND DISCUSSION

8.1 Web Application

8.1.1 AboutUs

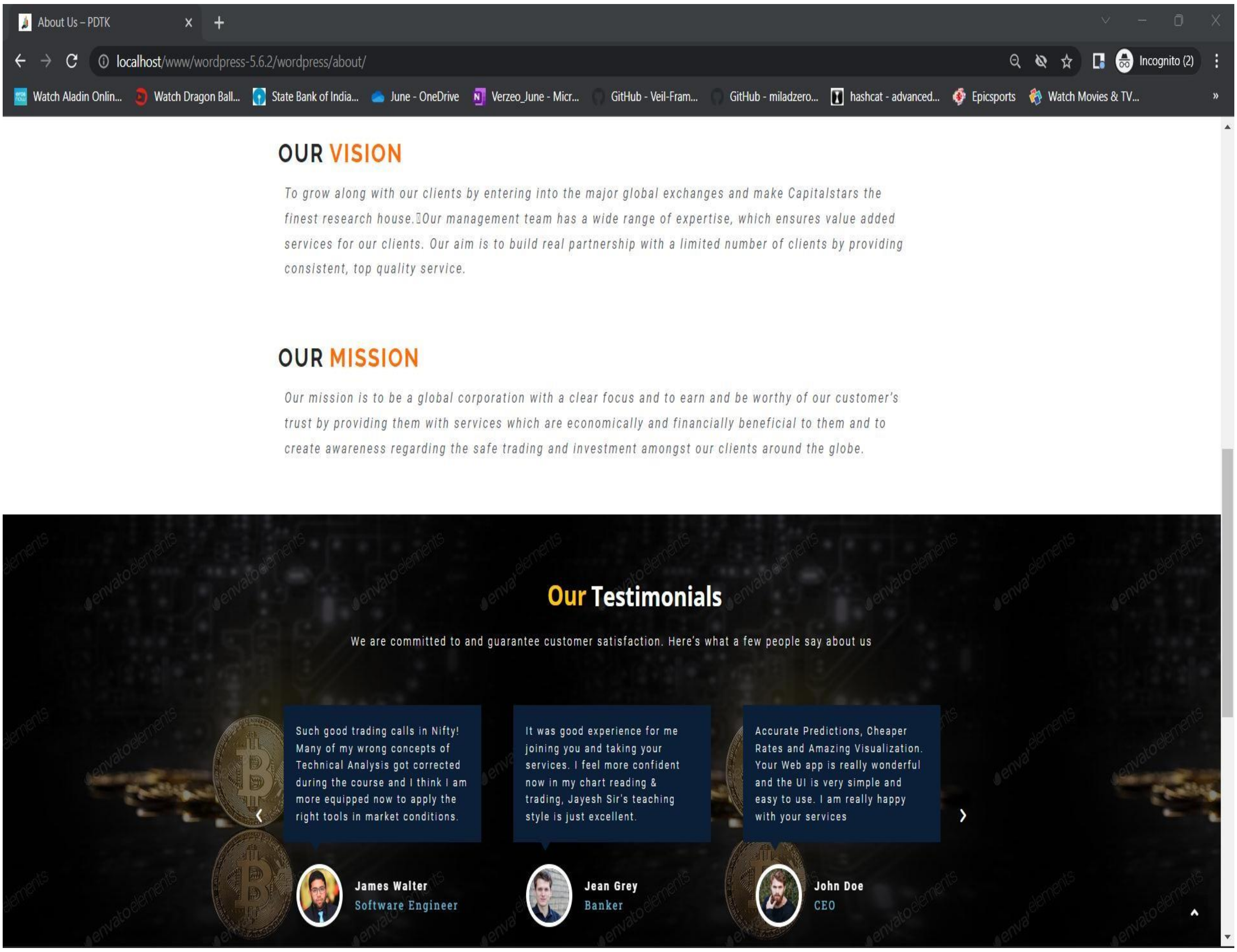


Fig:8.1.1.1 About Us

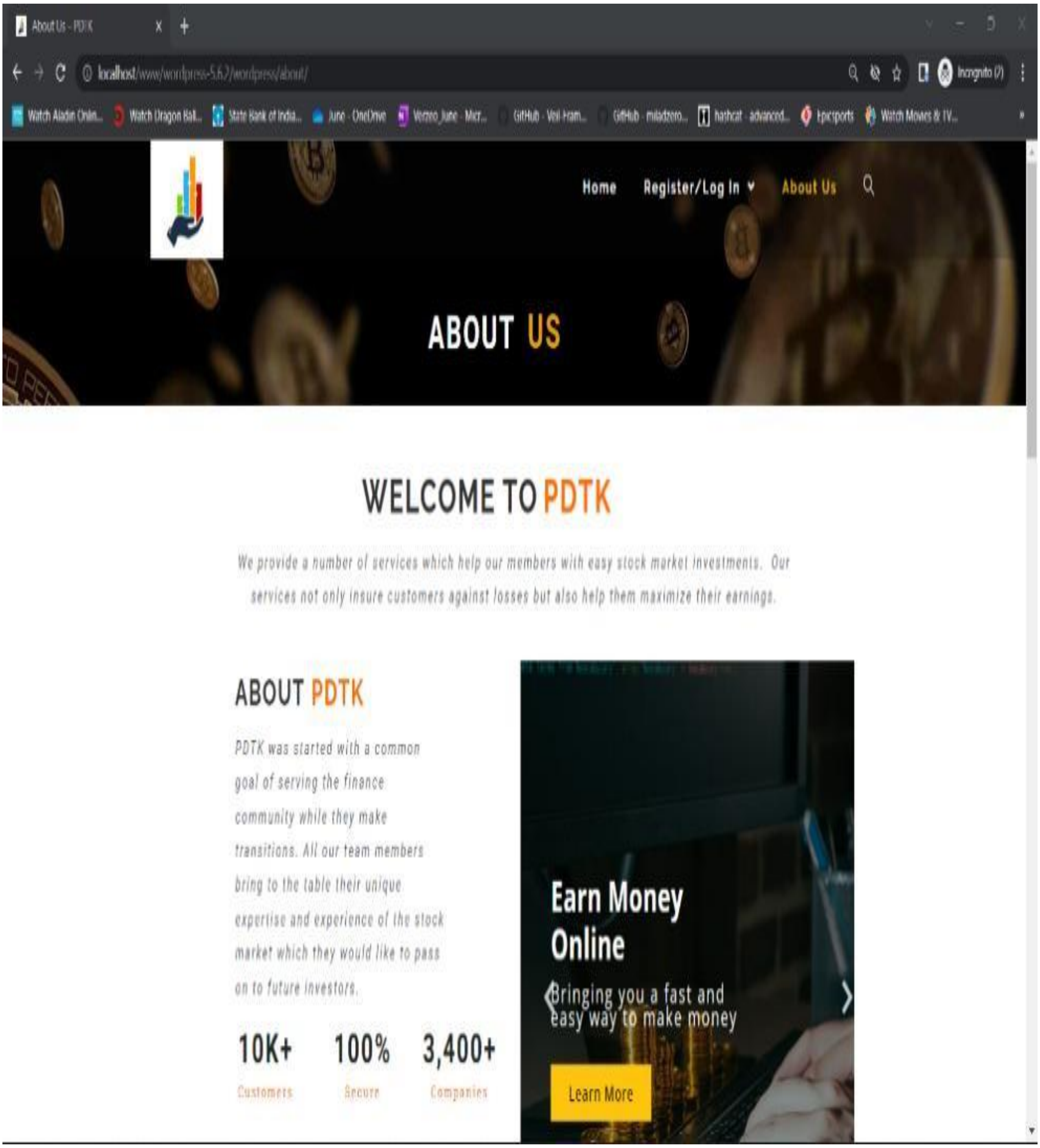


Fig:8.1.1.2 About Us

8.1.2 After user logged in

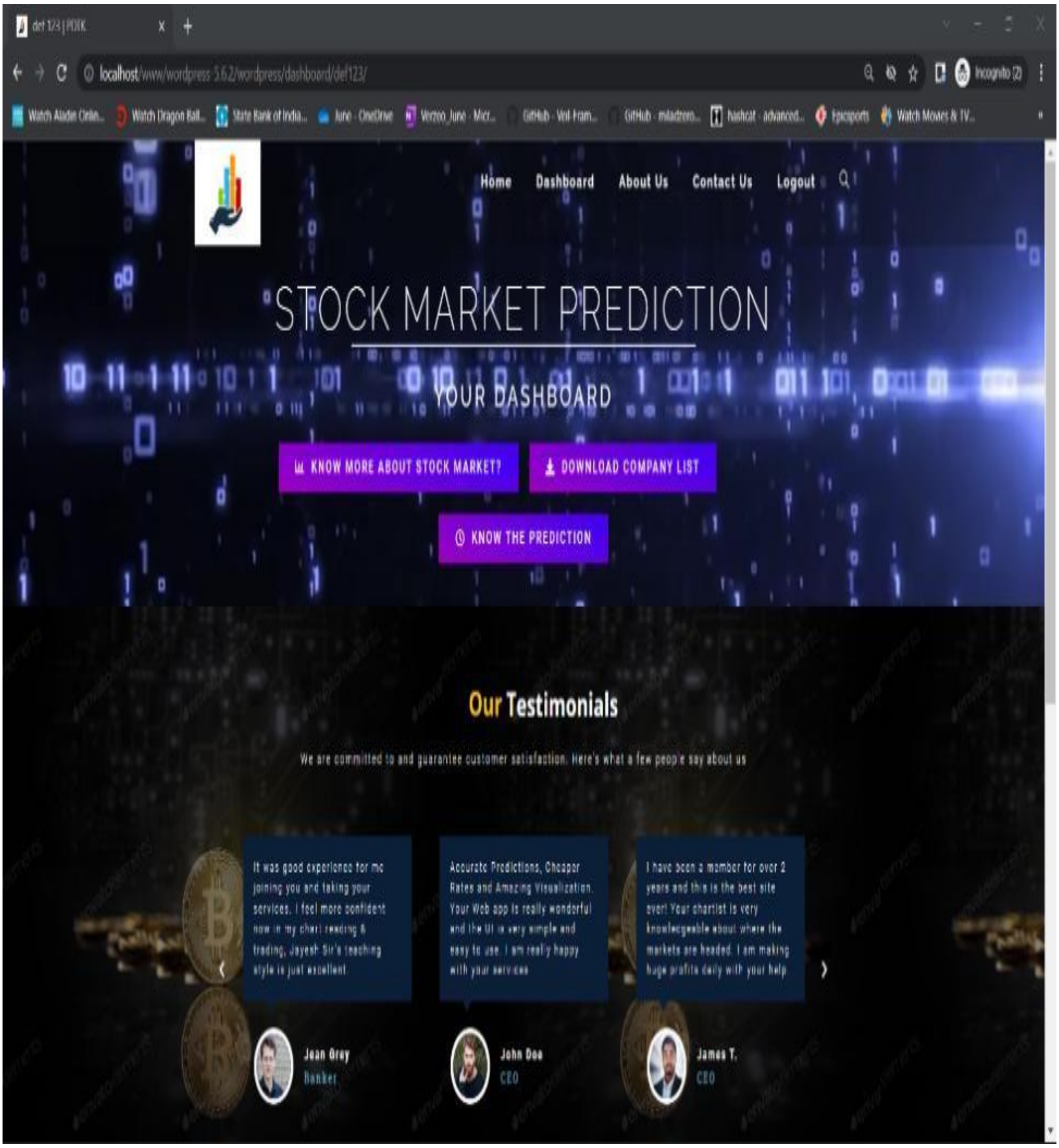


Fig: 8.1.2.1 After user logged in

8.1.3 Home page

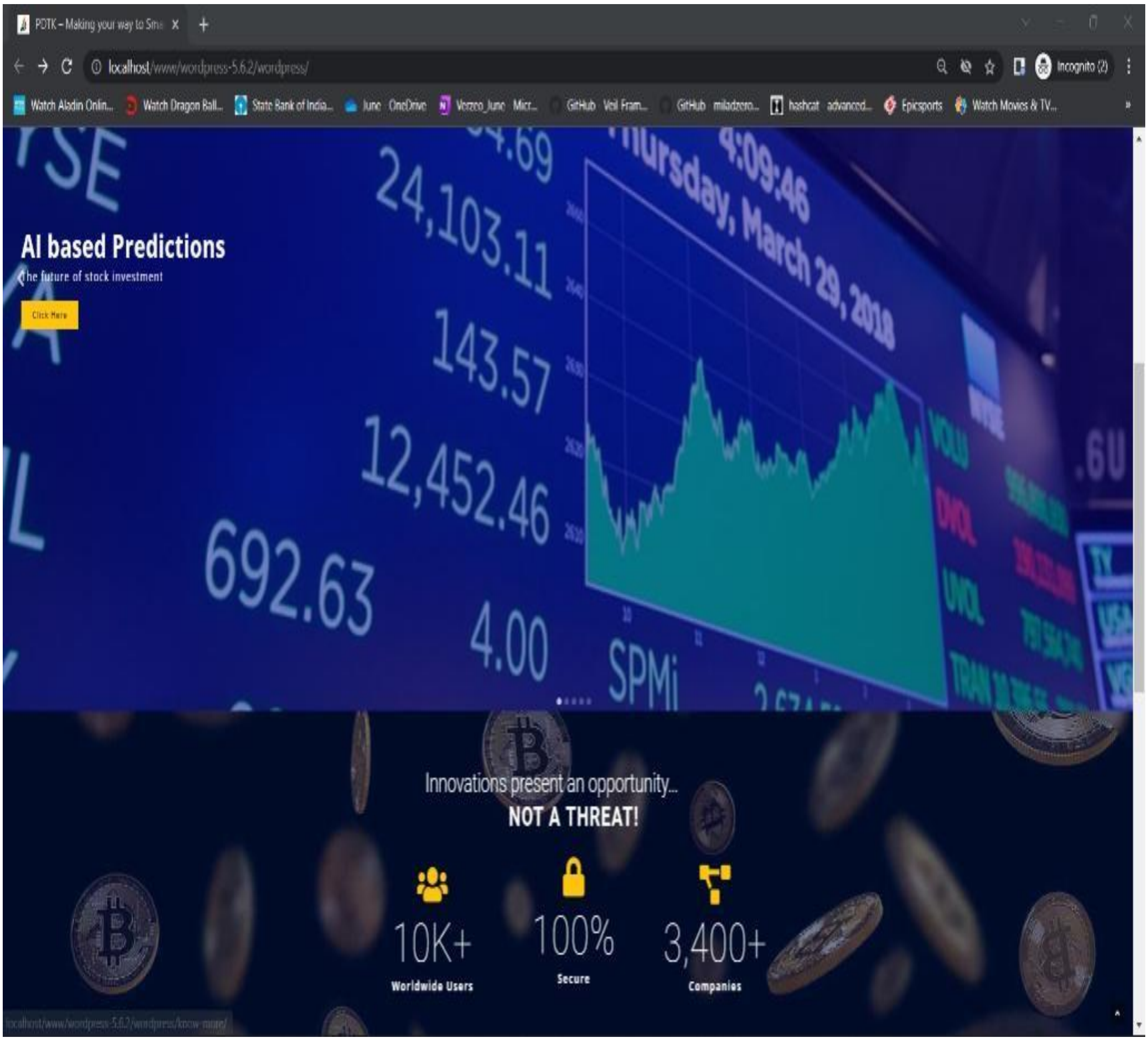


Fig: 8.1.3.1 Home page

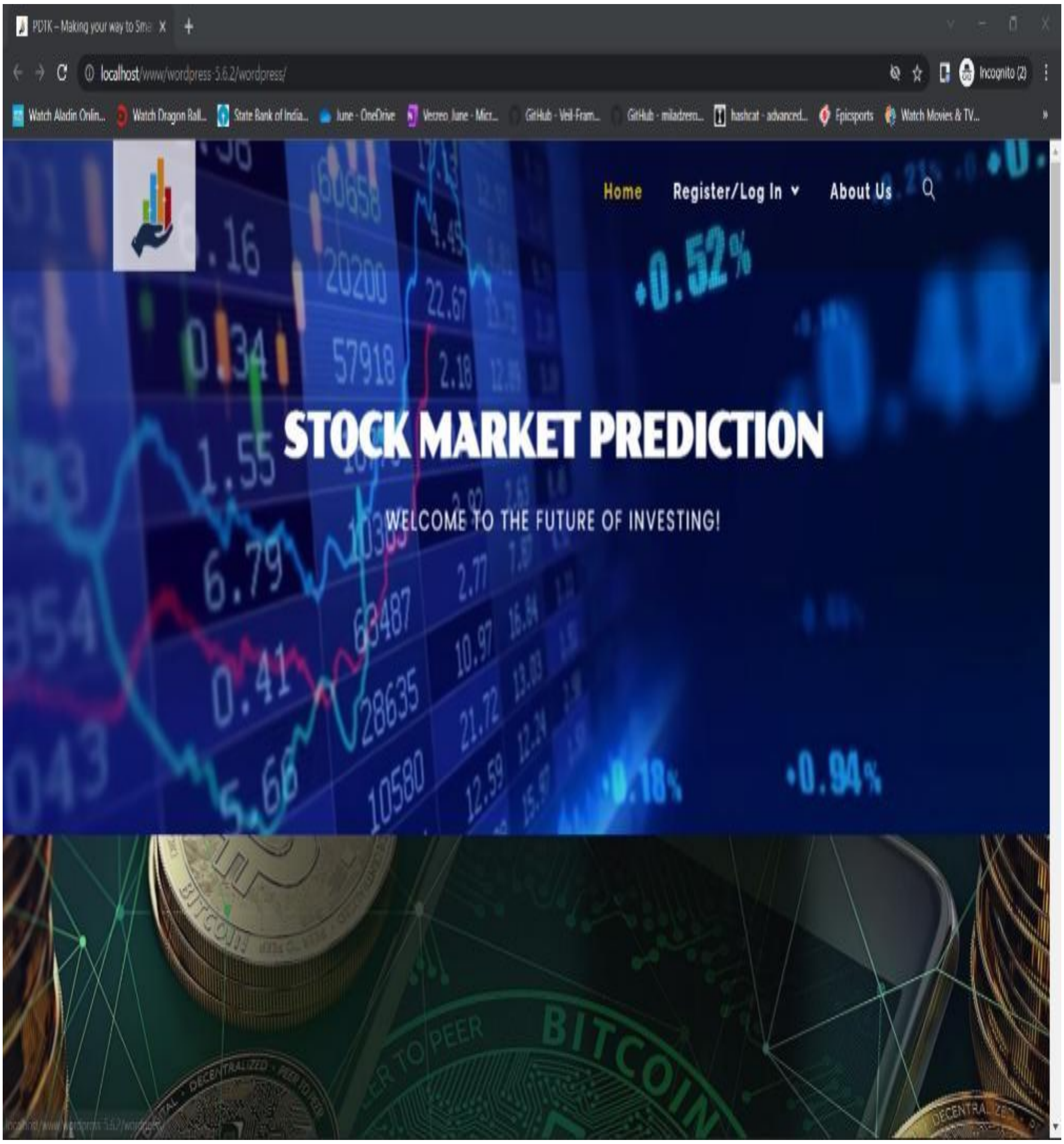


Fig: 8.1.3.2 Home page

8.1.4 Know more

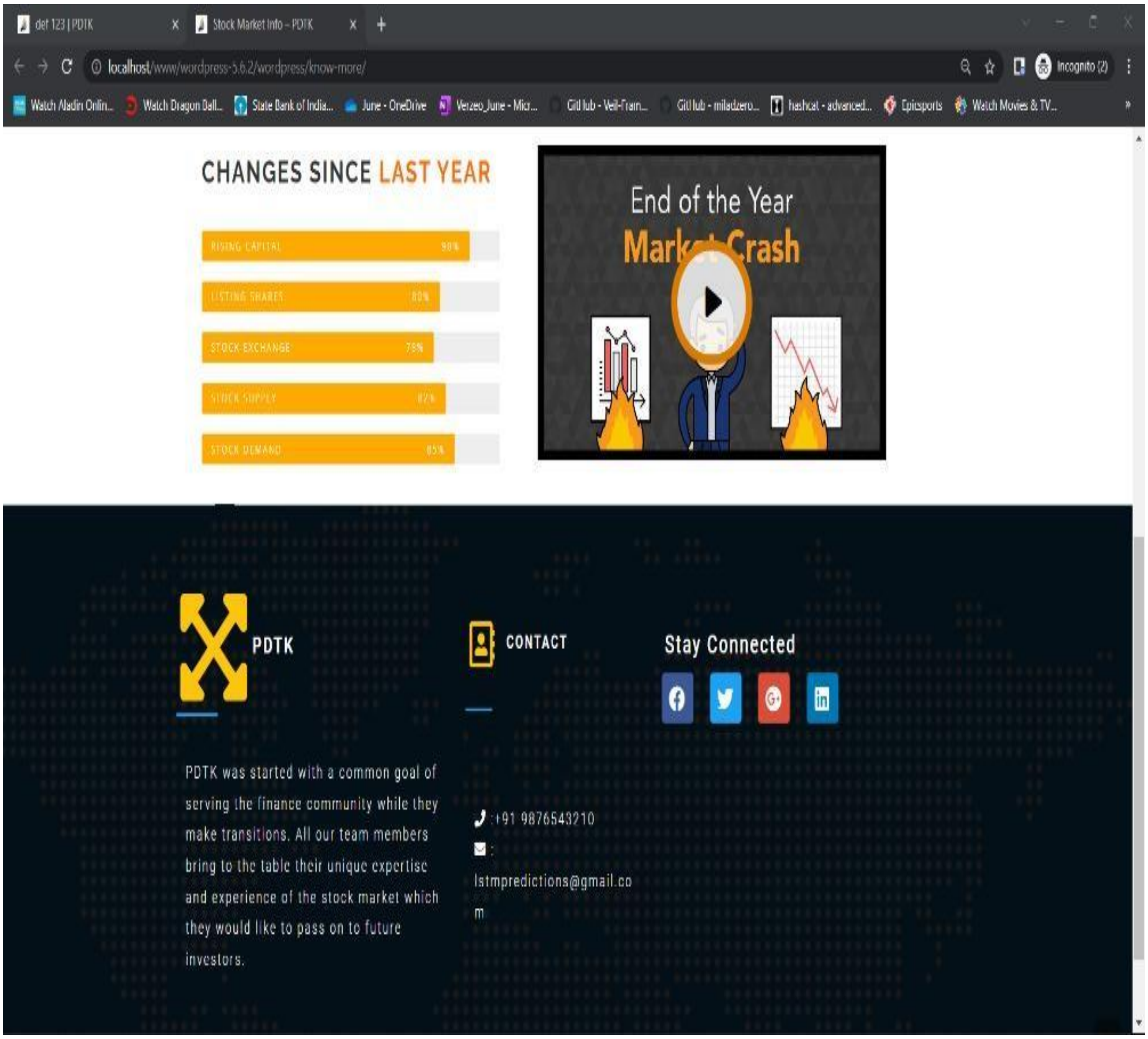


Fig: 8.1.4.1 Know more

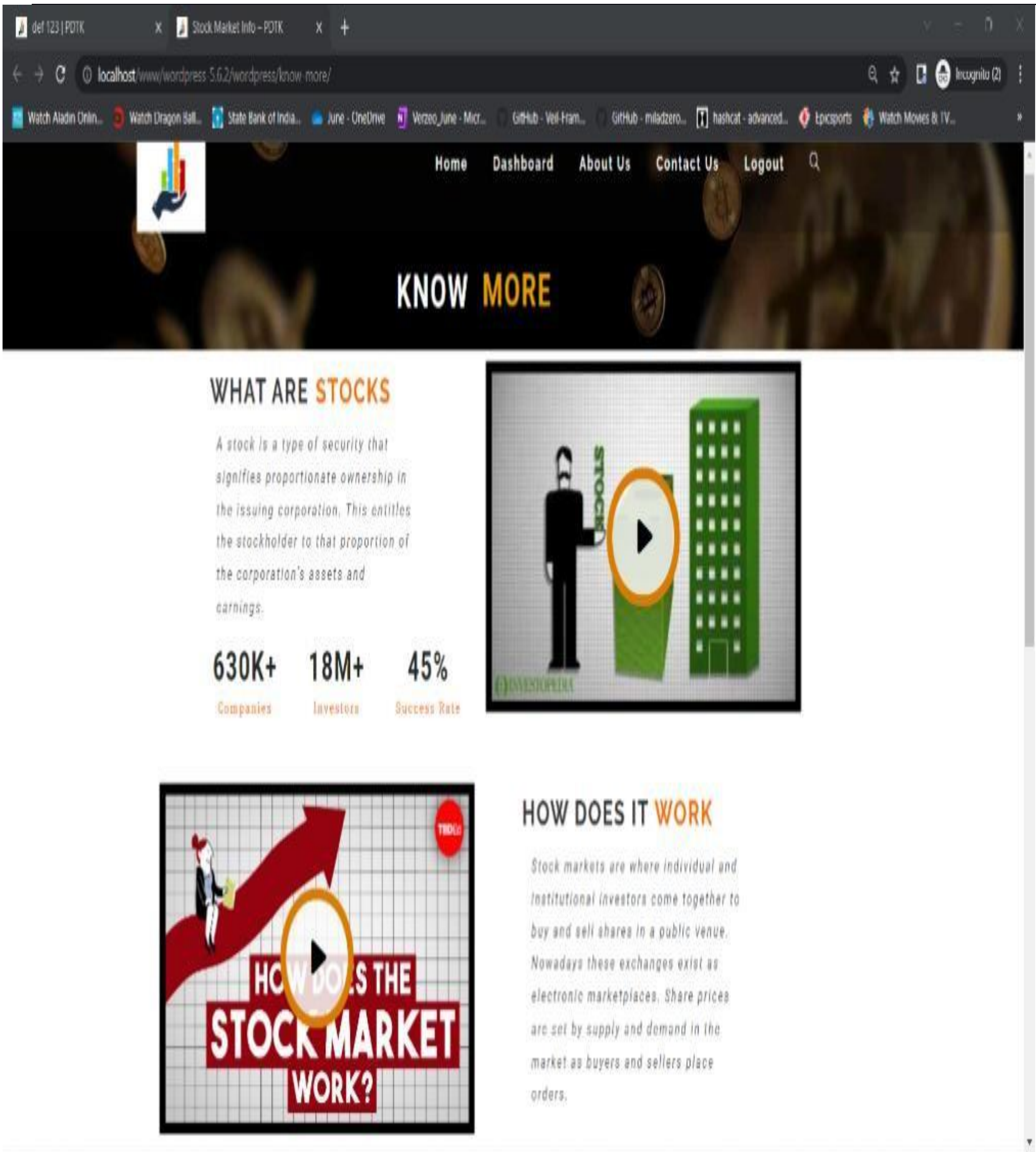


Fig: 8.1.4.2 know more

8.1.5 login page

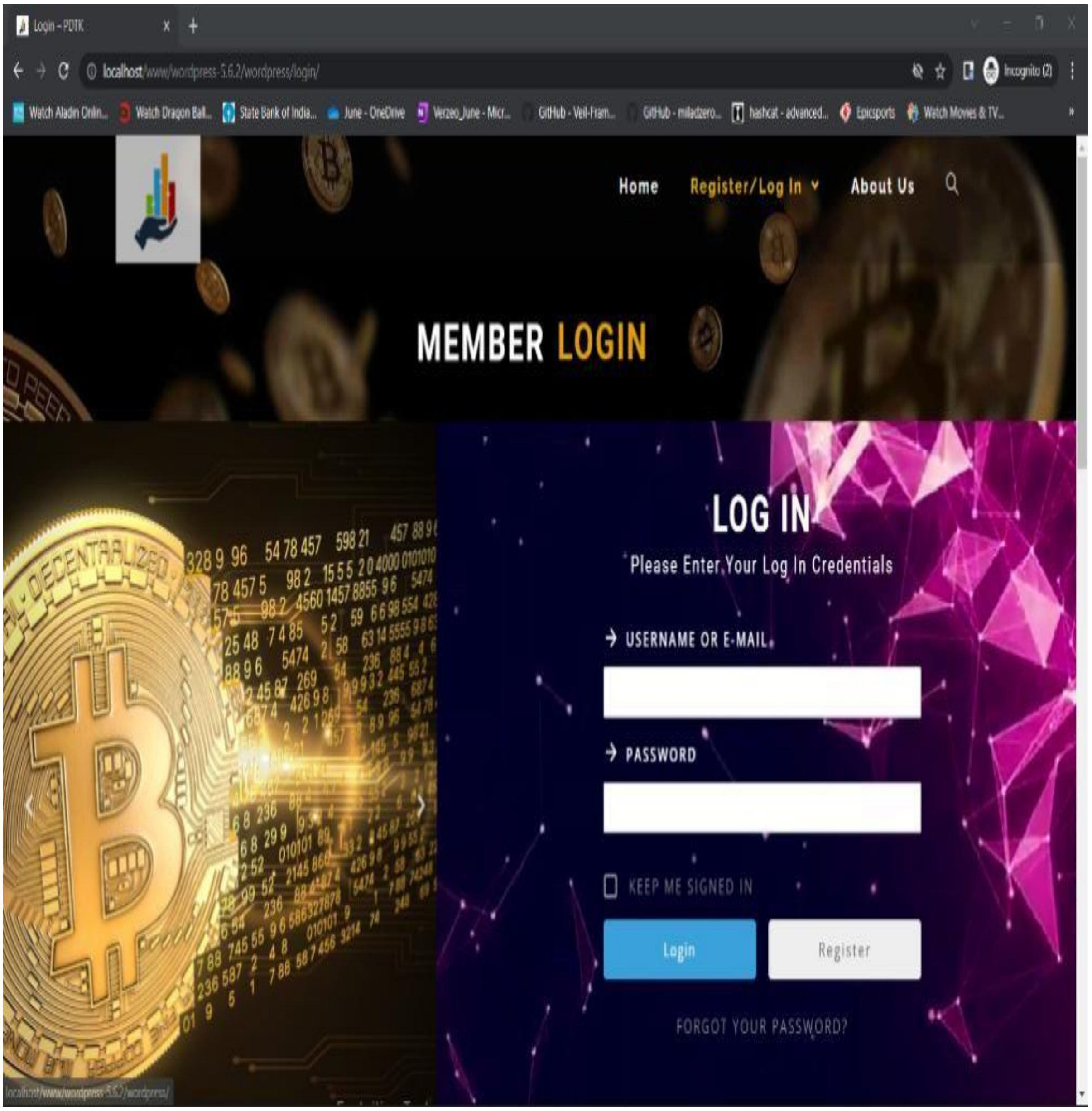


Fig: 8.1.5.1 login page

8.1.6 Prediction

Fig: 8.1.6.1 Prediction

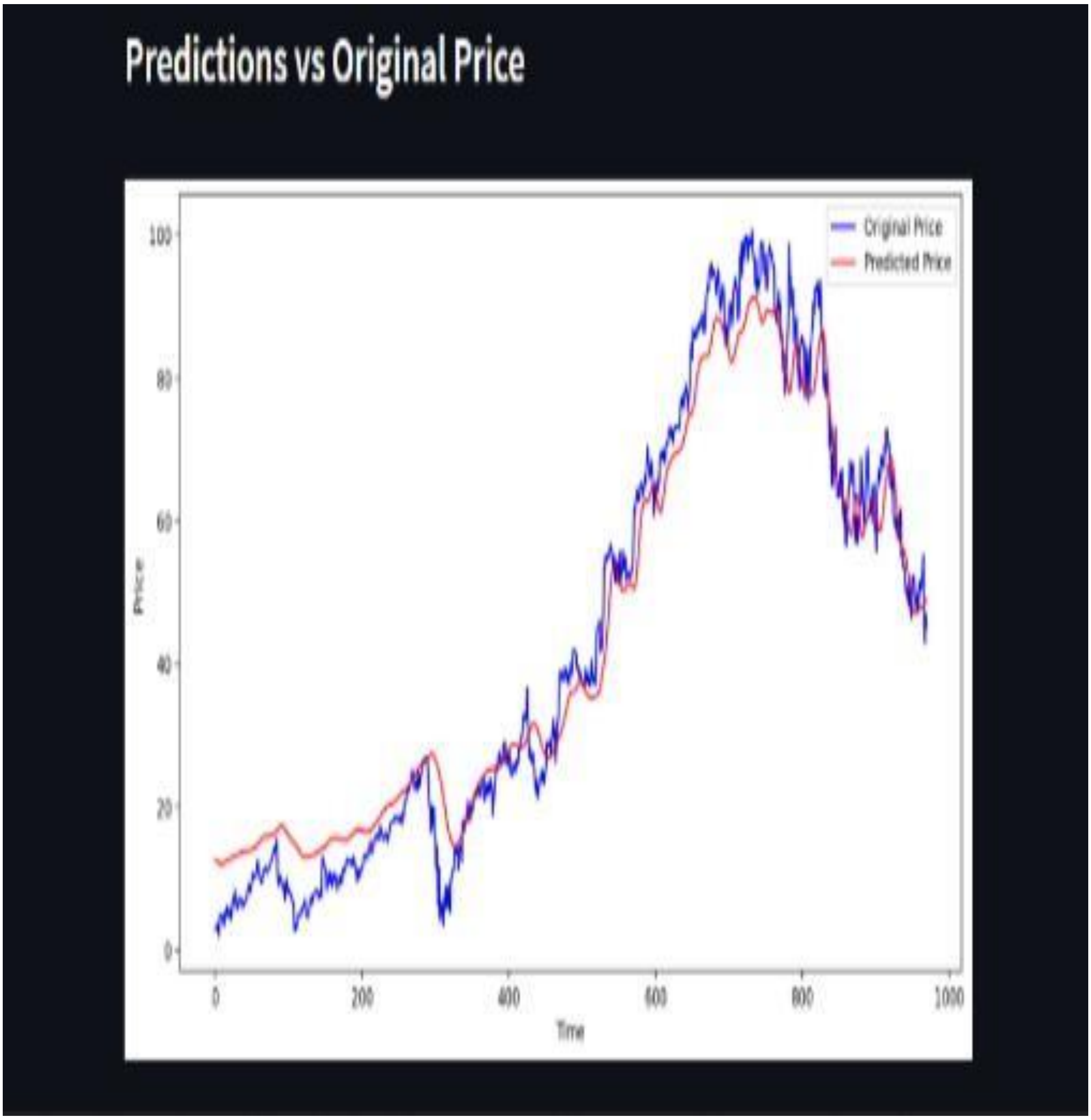


Fig: 8.1.6.2 Prediction

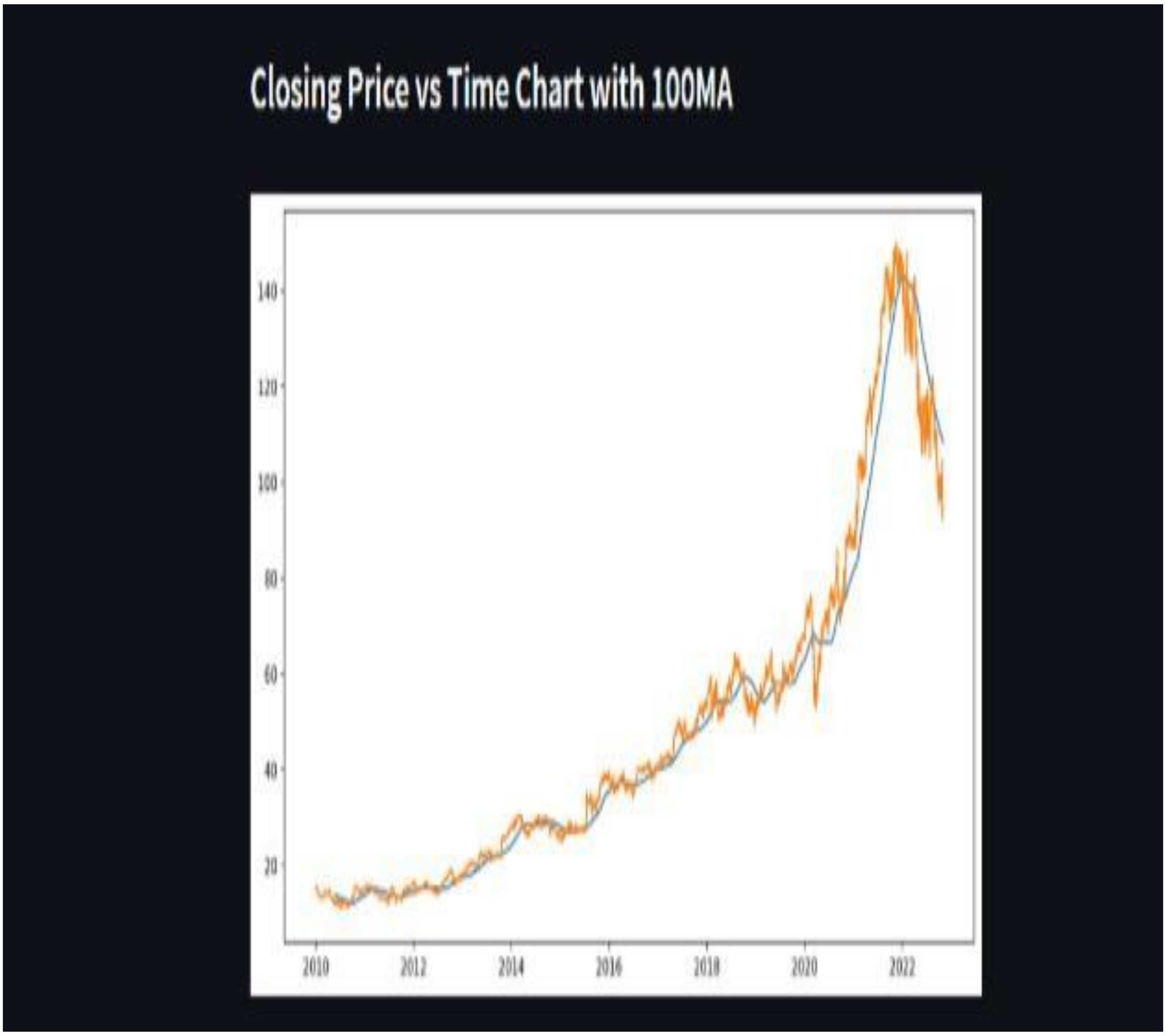


Fig: 8.1.6.3 Prediction

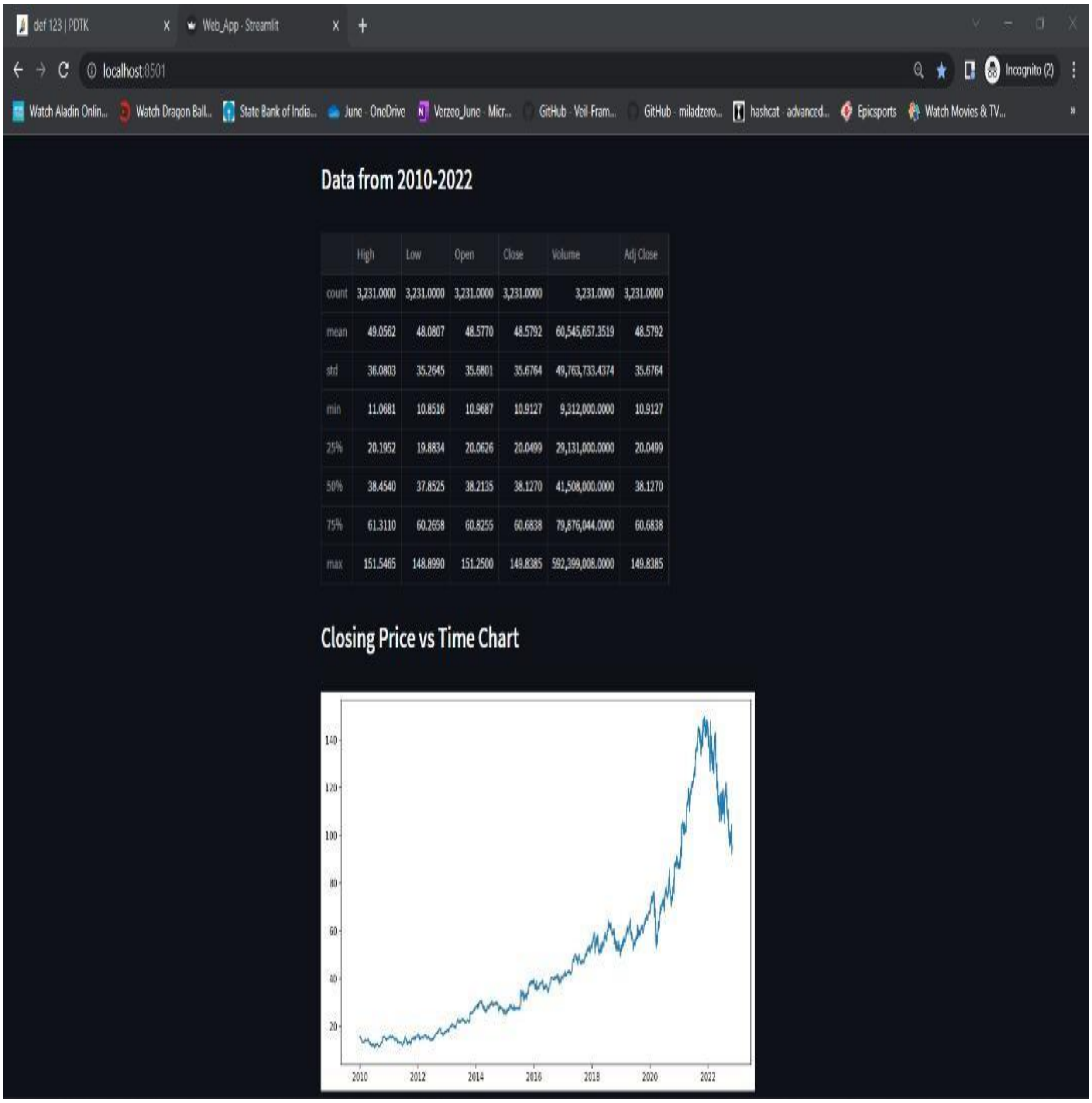


Fig: 8.1.6.4 Prediction

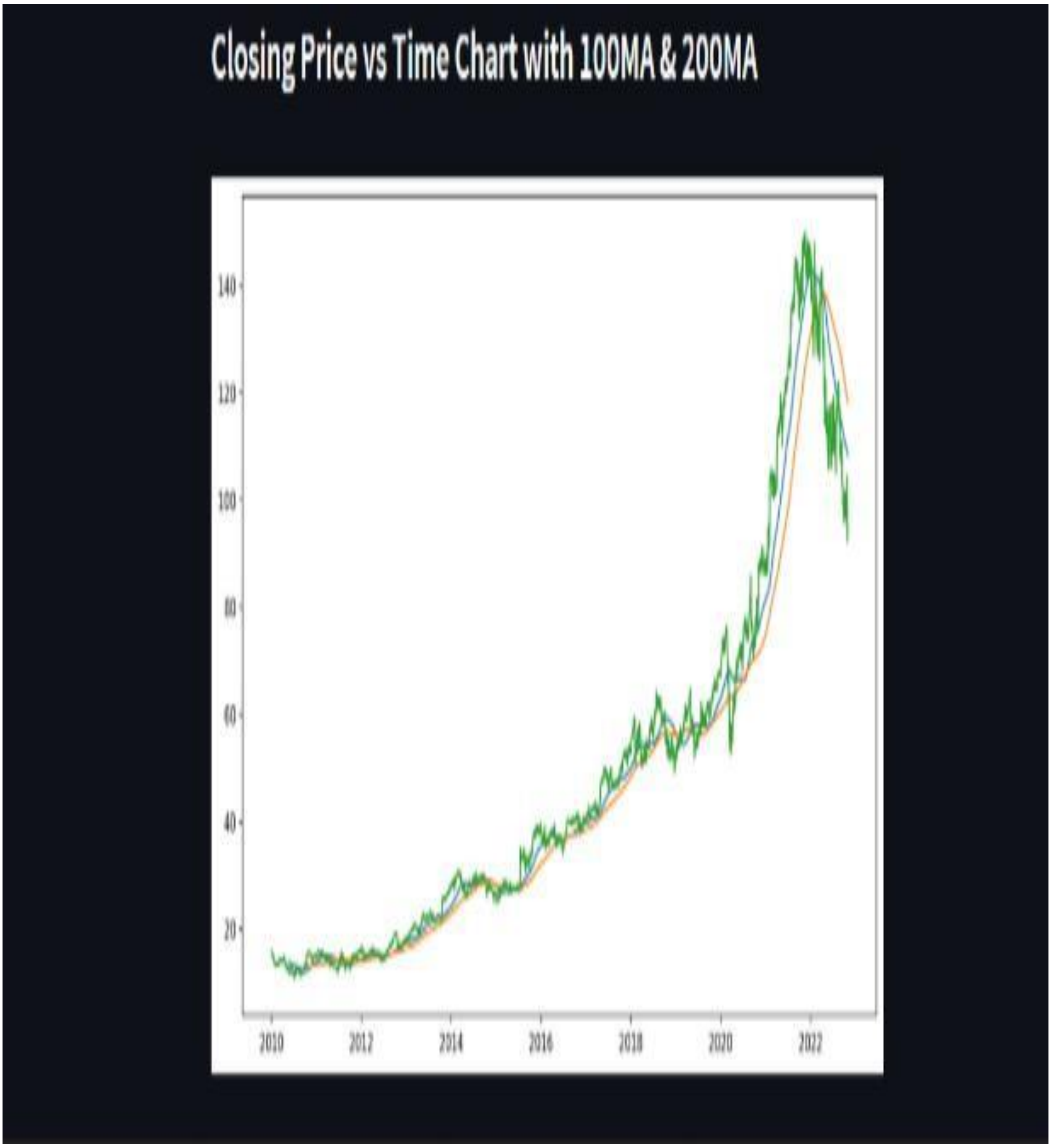


Fig: 8.1.6.5 Prediction

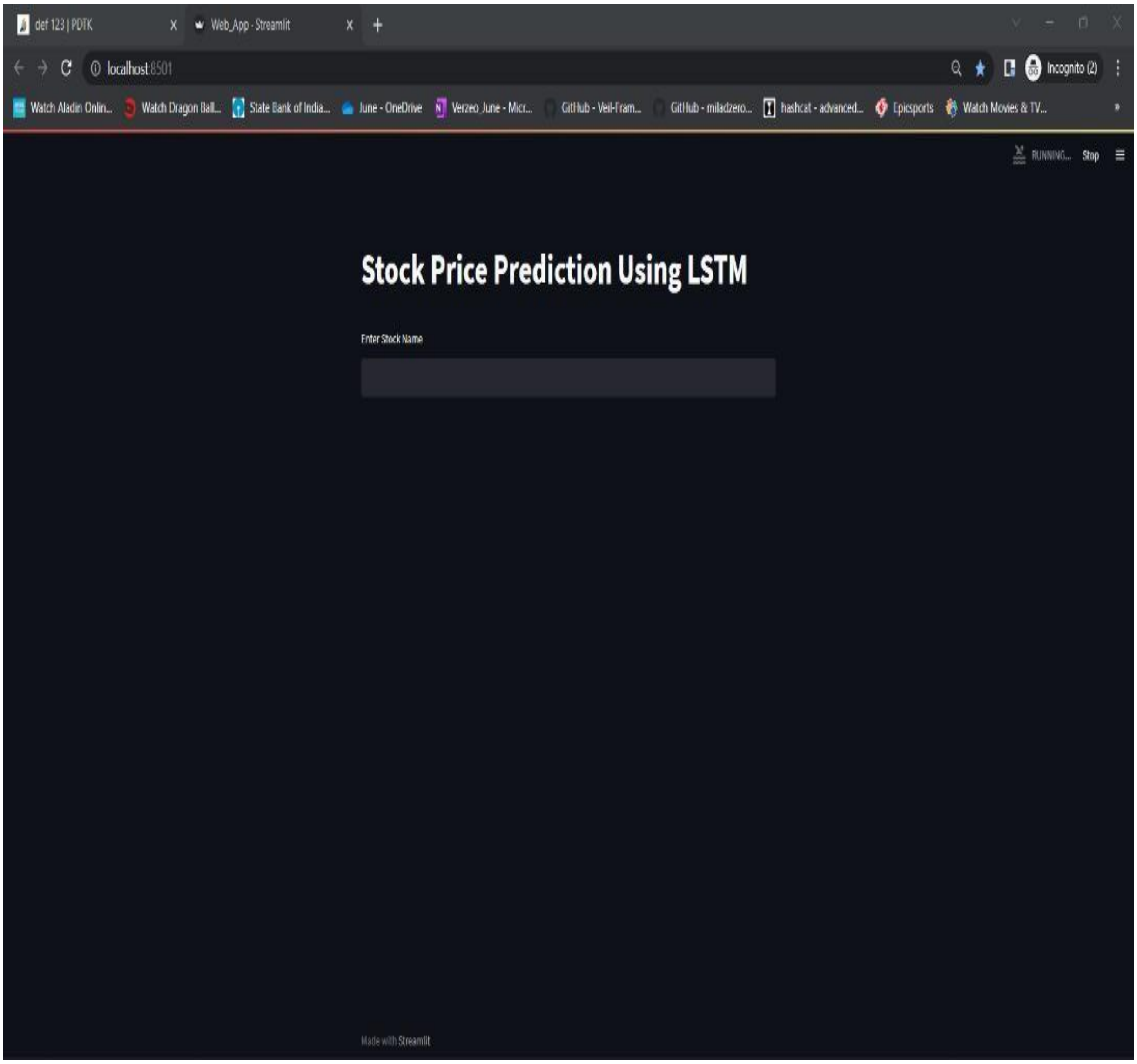


Fig:8.1.6.6 Prediction

8.1.7 Register page

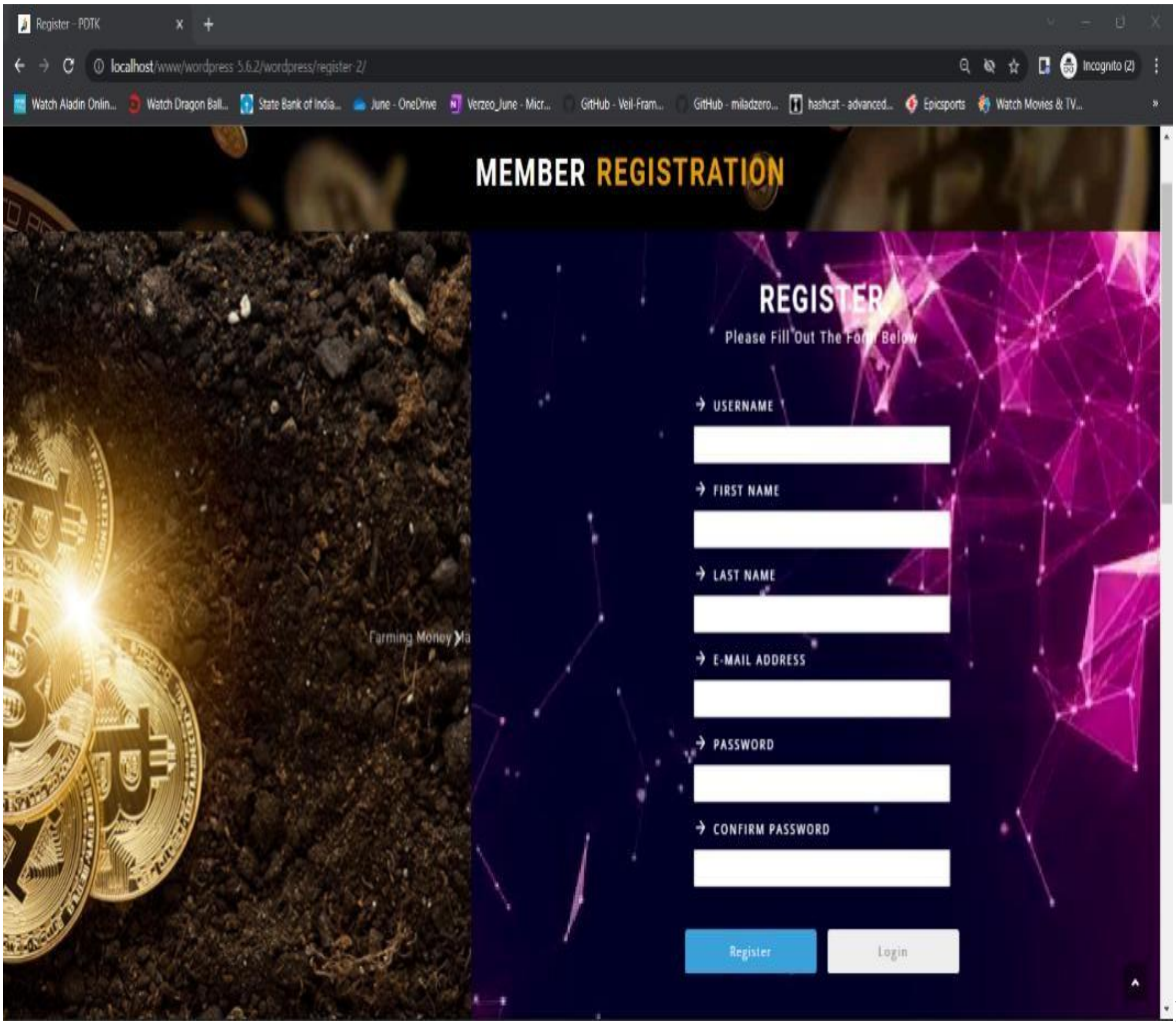


Fig: 8.1.7.1 Registration page

8.2 Final Output Graph

```
In [50]: plt.figure(figsize=(12,6))
         plt.plot(y_test,'b',label='Original Price')
         plt.plot(y_predicted,'r',label='Predicted Price')
         plt.xlabel('Time')
         plt.ylabel('Price')
         plt.legend()
         plt.show()
```

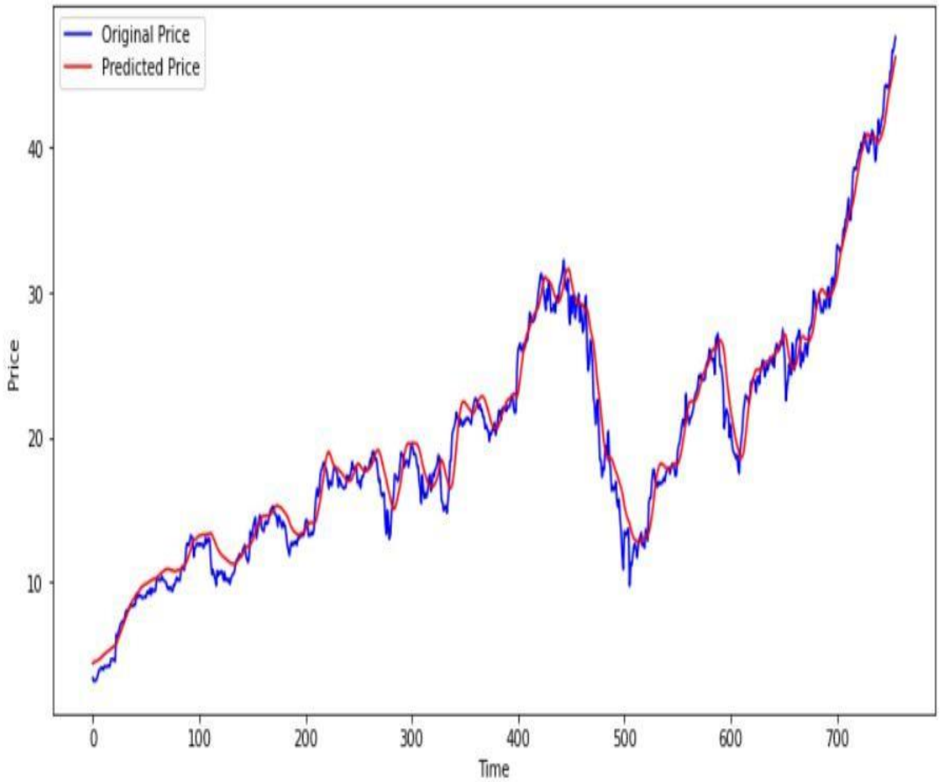


Fig: 8.2.1 Final Output Graph

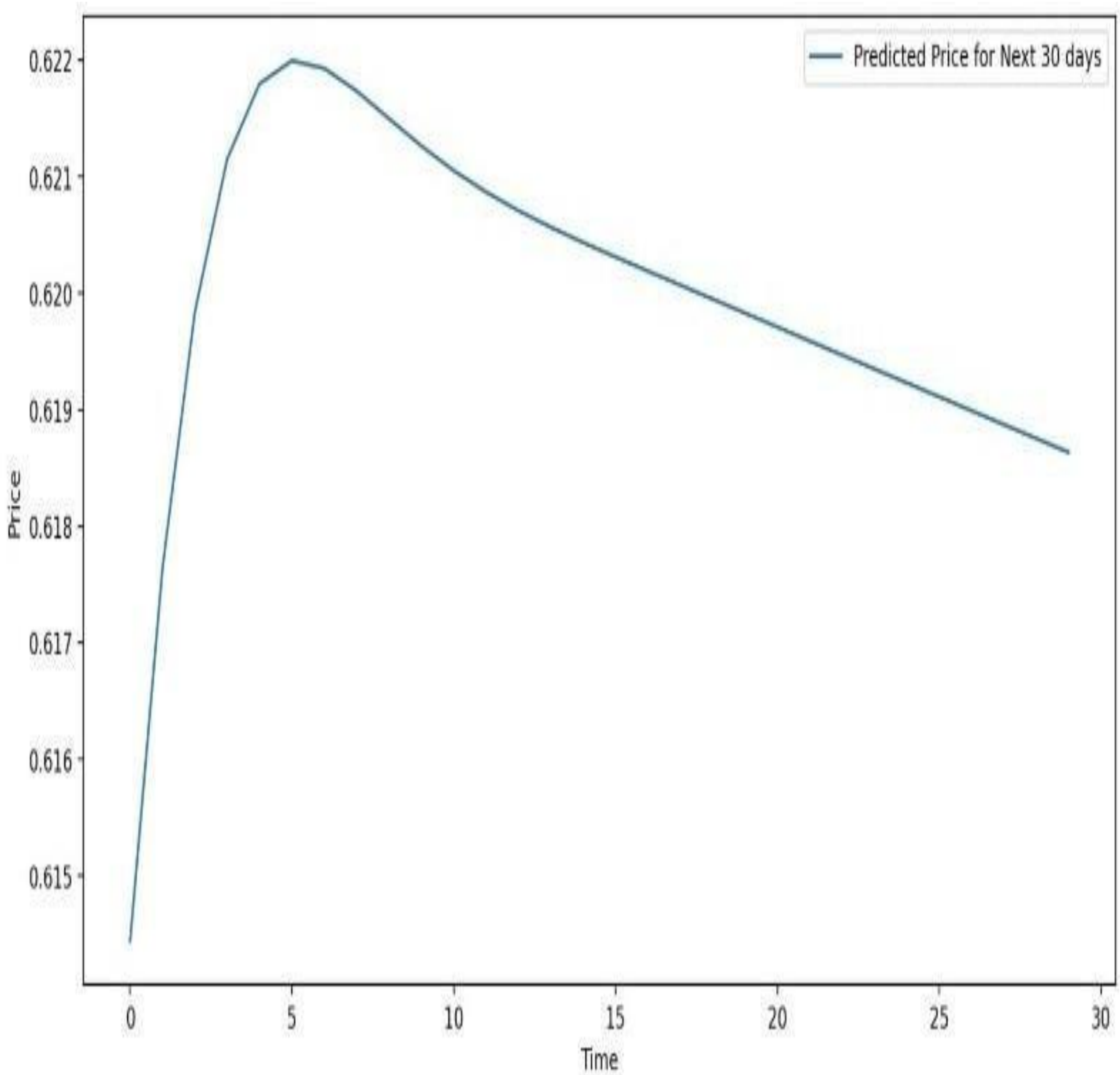


Fig: 8.2.2 Final Output Graph

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Conclusion

- We have splitted our data into training and testing and we modelled the data using LSTM model at last we plotted the graph of actual and predicted stock prices full stock
- We designed a web application to predict stock prices that can register and store user information within the database and allows us to select stock from company based on interest

9.2 Future work

- We would try to implement cryptocurrency trading option in the website.
- We also would like to try adding sentinel analysis in the algorithm.

REFERENCES/BIBLIOGRAPHY

- Stock Price Prediction Using LSTM on Indian Share Market by Achyut Ghosh, Soumik Bose1, Giridhar Maji, Narayan C. Debnath, Soumya Sen-2012
- S. Selvin, R. Vinaya kumar, E. A. Gopalkrishnan, V. K. Menon and K. P. Soman - Stock price prediction using LSTM, RNN and CNN-sliding window model - 2017
- Murtaza Roondiwala, Harshal Patel, Shraddha Varma, "Predicting Stock Prices Using LSTM" in Undergraduate Engineering Students, Department of Information Technology, Mumbai University,2005
- Osman Hegazy, Omar S. Soliman and Mustafa Abdul Salam Faculty of Computers and • Informatics, Cairo University, Egypt - A Machine Learning Model for Stock Market Prediction 2015 • M. Mekayel Anik · M. Shamsul Arefin (B) Department of Computer Science and Engineering, Chittagong University of Engineering and Technology, Chittagong, Bangladesh - An Intelligent Technique for Stock Market Prediction.
- Selvin ,Kumar research paper - Workload forecasting with Long short term memory" (2017) (2018)
- Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan Department of Computer Science and Engineering National Institute of Technology, Hamirpur – 177005, INDIA - Stock Market Prediction Using Machine Learning.
- Pranav Bhat Electronics and Telecommunication Department, Maharashtra Institute of Technology, Pune. Savitribai Phule Pune University - A Machine Learning Model for Stock Market Prediction. [7] Anurag Sinha Department of computer science, Student, Amity University Jharkhand Ranchi, Jharkhand (India), 834001 - Stock Market Prediction Using Machine Learning.
- V Kranthi Sai Reddy Student, ECM, Sreenidhi Institute of Science and Technology, Hyderabad, India - Stock Market Prediction Using Machine Learning.

- Sharanya Banerjee, Neha Dabeeru, R. Lavanya SRM Institute of Science and Technology, Chennai in Computer Science and Engineering - Stock Market Prediction
- Fahad Almudhaf, Yaser A. Alkulaib from Kuwait University - Are Civets Stock Markets Predictable? [23] Mohammad Mekayel Anik, Mohammad Shamsul Arefin and M. Ali Akber Dewan, Department of Computer Science and Engineering - An Intelligent Technique for Stock Market Prediction
- Asep Juarna, Departemen of Informatics, Gunadarma University, Jakarta-Indonesia - ONE YEAR STOCK PRICE PREDICTION AND ITS VALIDITY USING LEAST SQUARE METHOD IN MATLAB
- Giovanis, Eleftherios - Applications of Least Mean Square (LMS) Algorithm Regression in TimeSeriesAnalysis
- Vivek Rajput and Sarika Bobde - Stock market prediction using hybrid approach
- James C. Van Horne and George G. C. Parker – The Random-walk Theory: An Empirical Test
- Asset Durmagambetov currently works at the mathematics, CNTFI- 'The Riemann HypothesisMillennium Prize Problems' - stock market predictions.
- Mariam Moukalled Wassim El-Hajj Mohamad Jaber Computer Science Department American University of Beirut - Automated Stock Price Prediction Using Machine Learning.
- Manh Ha Duong Boriss Siliverstovs June 2006 - The Stock Market and Investment.
- Dharmaraja Selvamuthu, Vineet Kumar and Abhishek Mishra Department of Mathematics, Indian Institute of Technology Delhi, Hauz Khas, New Delhi 110016, India - Indian stock market prediction using artificial neural networks on tick data.
- Lufuno Ronald Marwala A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering - Forecasting the Stock Market Index Using Artificial Intelligence Techniques.

- Xiao Ding, Kuo Liao, Ting Liu, Zhongyang Li, Junwen Duan Research Centre for Social Computing and Information Retrieval Harbin Institute of Technology, China - Event Representation Learning Enhanced with External Common-sense Knowledge.
- Huicheng Liu Department of Electrical and Computer Engineering Queen's University, Canada - Leveraging Financial News for Stock Trend Prediction with Attention-Based Recurrent Neural Network
- Hyeong Kyu Choi, B.A Student Dept. of Business Administration Korea University Seoul, Korea = Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model.
- M. Nabipour Faculty of Mechanical Engineering, Tarbiat Modares University, 14115-143 Tehran, Iran;Mojtaba.nabipour@modares.ac.ir - Deep Learning for Stock Market prediction.
- Lavanya Ra SRM Institute of Science and Technology | SRM · Department of Computer Science - Stock Market Prediction