ST&P: A Stock Tracker & Predictor

A Project Submitted

In partial fulfillment of the requirement

for the award of

Master of Computer Applications

by:

Abhishek Gupta	2021CA007
Dheeraj Rajput	2021CA030
Manish Vardhan	2021CA060
Palpal Kerai	2021CA074
Hasib Khan	2021CA043

under the guidance of

Dr. Pragya Dwivedi

to the

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY ALLAHABAD

May 2023

UNDERTAKING

We declare that the work presented in this report titled **"ST&P: A Stock Tracker & Predictor"** submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, for the award of the Master of Computer Applications degree, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, we accept that our degree may be unconditionally withdrawn.

May 2023

Allahabad

_____

Abhishek Gupta

Dheeraj Rajput

Manish Vardhan

Palpal Kerai

Hasib Khan

CERTIFICATE

This is to certify that Abhishek Gupta, Dheeraj Rajput, Manish Vardhan, Palpal Kerai, and Hasib Khan have successfully carried out the completion of the project entitled **"ST&P: A Stock Tracker & Predictor"** under my supervision during session 2022-23 and all the requirements of the project have been met.

May 2023

_____

Dr. Pragya Dwivedi

Computer Science and Engineering Dept.

MNNIT Allahabad, Prayagraj

ACKNOWLEDGMENT

We would like to give our sincere thanks to our mentor, **Dr. Pragya Dwivedi,** who guided us throughout our project **"ST&P: A Stock Tracker & Predictor"** in every possible way with her invaluable advice and valuable suggestions. Our project would not have been possible without her constant guidance, support, and encouragement that helped us complete our project on time. We feel honored and privileged to work under her.

Abhishek Gupta

Dheeraj Rajput

Manish Vardhan

Palpal Kerai

Hasib Khan

Contents

Chapter 1

About this project:

The Stock Tracker and Predictor Website is a web-based platform that provides users with real-time information on the stock market and help them make informed investment decisions. This project developed to provide users with a user-friendly and intuitive platform to track the performance of stocks and predict future stock prices.

The website's key features include real-time stock price updates, historical price data, and prediction algorithms. The website's prediction algorithms developed using machine learning and artificial intelligence techniques, which allow for accurate and reliable predictions of future stock prices.

Overall, this project represents an innovative and proper application of technology to the finance industry. By providing users with real-time stock market data and predictive analytics, this website has the potential to revolutionize the way investors and traders approach the stock market.

Chapter 2

Introduction

1. Motivation

The motivation behind the Stock Tracker and Predictor Website project is to provide investors, traders, and finance professionals with a user-friendly and intuitive platform to track the performance of stocks and predict future stock prices based on various algorithms.

2. Objective

The objective of the Stock Tracker and Predictor Website is to provide users with a user-friendly and intuitive platform to track the performance of stocks and predict future stock prices based on various algorithms. The website aims to provide users with accurate and reliable information on the stock market and empower them to make better investment decisions.

In this report, we will provide a detailed analysis of the Stock Tracker and Predictor Website, including its features, architecture, and implementation details. We will also discuss the benefits of using this website and how it can help users make more informed investment decisions. Finally, we will provide some recommendations for future improvements to the website.

Chapter 3

Proposed Work

1. Requirement gathering: This involves identifying the essential requirements and features of the website, such as real-time stock price updates, historical price data, prediction algorithms, and a user-friendly interface.

2. Design and architecture: This involves designing the website's front-end, back-end, and database architecture. The front-end design should be intuitive and user-friendly, while the back-end should be robust and scalable to handle large volumes of data.

3. Development involves implementing the design and architecture using React [1], Tailwind CSS [6], and Node.js [2]. The website's front end should be responsive, ensuring that it is optimized for different devices, including desktops, tablets, and mobile devices.

4. Testing: This involves testing the website's functionality, performance, and security to ensure that it meets the project's requirements and standards.

5. Deployment: This involves deploying the website to a production environment, making it accessible to users.

6. Maintenance and updates: This involves monitoring the website's performance, fixing bugs, and updating the website's features and functionality to ensure that it remains relevant and up-to-date.

3.1 Data Collection

Collecting relevant data is one of the most important aspects of building a stock tracker and predictor website. The website will require real-time stock market and historical price data for stocks to provide users with accurate and reliable information.

The real-time stock market data obtained from various sources such as APIs provided by stock exchanges, financial news outlets, and financial data providers such as Yahoo Finance, Stock NSE Index.

3.2 Data Pre-processing:

Data pre-processing is a critical step in building a stock tracker and predictor website, as it involves cleaning and transforming the data to ensure that it is accurate and reliable. The following are the common data pre-processing steps:

1. **Data Cleaning**: This step involves identifying and removing or correcting any errors, missing data, or outliers in the data. For example, if the data includes any missing or null values, we need to decide on how to handle them, whether to remove them or fill them with a specific value.

2. **Data Transformation**: This step involves converting the data into a suitable format for prediction algorithms. For example, we can normalize the data to a specific range, convert categorical variables into numerical variables, or use feature-scaling techniques.

3. **Feature Extraction**: This step involves selecting the relevant features that are most likely to influence the stock price. For example, we extracted features such as the opening and closing price, volume, and Adj Close.

4. **Data Reduction**: This step involves reducing the size of the dataset by removing redundant or irrelevant features, which can help in improving the performance of the prediction algorithms.

3.3 Prediction Model

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is widely used for stock price prediction. LSTMs are designed to overcome the limitations of traditional RNNs in modeling long-term dependencies in time-series data.

The key advantage of LSTMs over traditional RNNs is their ability to selectively remember or forget past inputs based on their relevance to the current output. This is achieved using gated cells, which control the flow of information through the network.

Here is a high-level overview of how an LSTM model used for stock price prediction:

1. Data preparation: The first step is to pre-process the stock price data to prepare it for the LSTM model. This typically involves normalization and splitting the data into training and testing sets.

2. Model architecture: The next step is to define the LSTM model architecture. The model typically consists of one or more LSTM layers followed by one or more dense layers.

3. Training the model: The next step is to train the LSTM model using the training data. During training, the model learns to predict the stock prices based on the historical data.

4. Model evaluation: Once the model is trained, it is evaluated using the testing data. The performance of the model is typically measured using metrics such as mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE).

5. Prediction: Once the model is trained and evaluated, it can be used to make predictions on new data. Given the historical data, the model predicts the future stock prices.

LSTMs have shown promising results in stock price prediction, but it is important to note that a wide range of factors, such as global economic conditions, political events, and news sentiment, affects stock prices. Therefore, it is important to consider other factors in addition to the LSTM model to make accurate stock price predictions.

Chapter 4

System Design

4.1 USE CASE DIAGRAM

A use case diagram is a visual representation of the functional requirements of a system, showing the interactions between the system and its users.

Here is a use case diagram for the Stock Tracker and Predictor website:

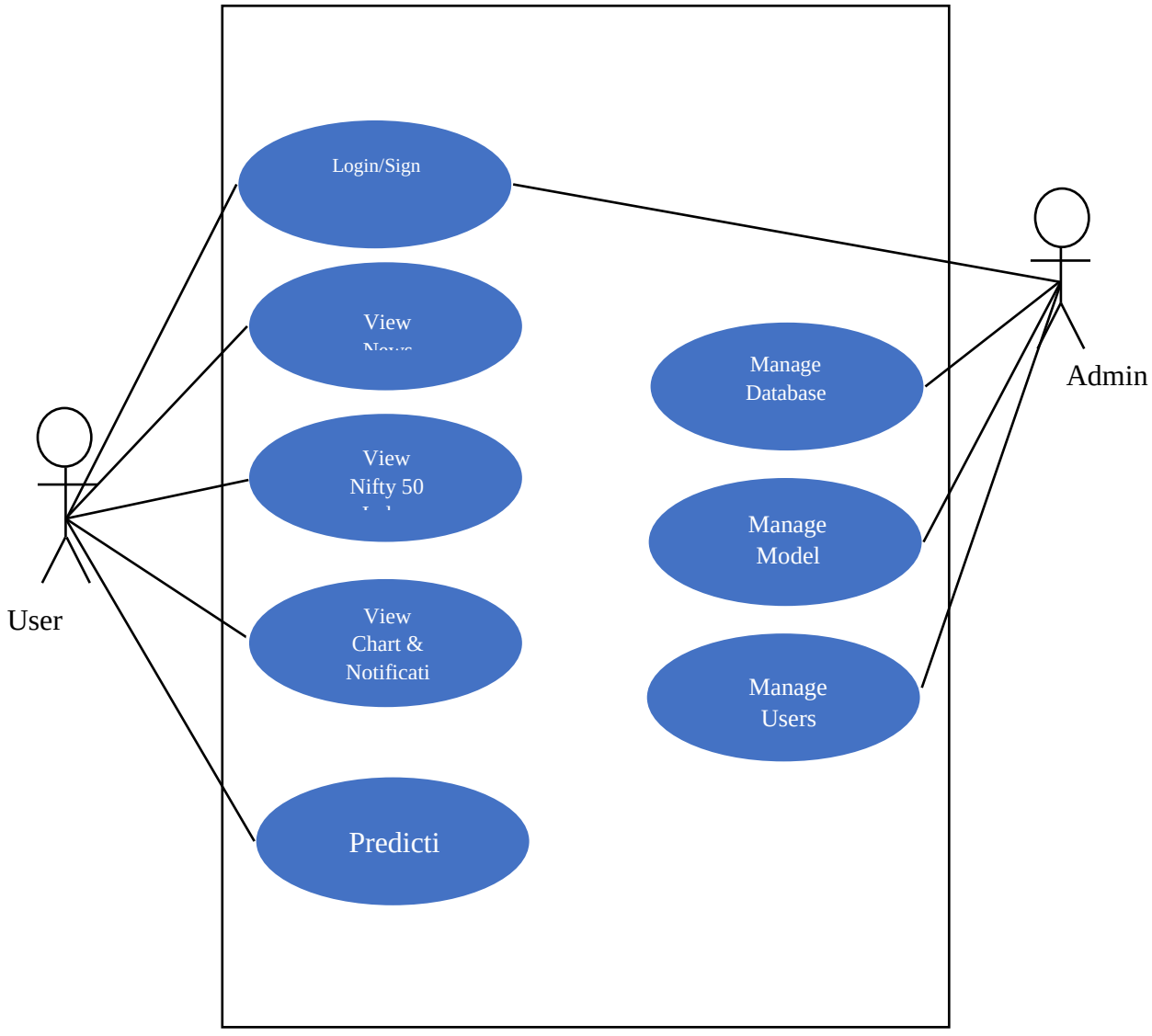


Fig 4.1 Use Case Diagram

4.2 ER Diagram

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system.

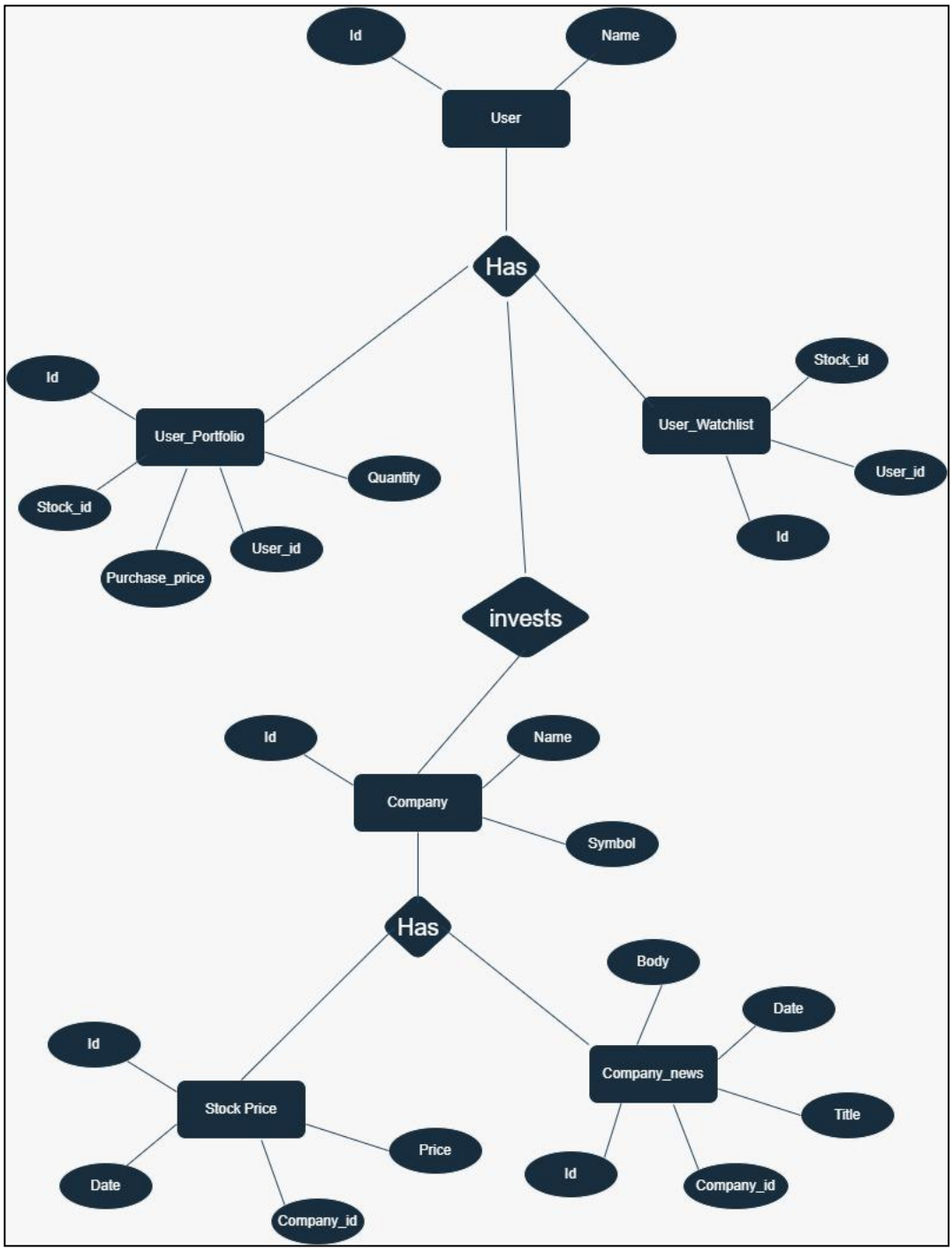


Fig 4.2 ER Diagram

4.3 DFD Diagram

DFD stands for Data Flow Diagram, which is a graphical representation of how data flows within a system.

There are different levels of DFDs, ranging from the high-level context diagram to detailed diagrams that show the internal workings of a system.

4.3.1 LEVEL 0:

DFD Level 0 is also called a Context Diagram. It is a basic overview of the whole system or process being analyzed or modeled.

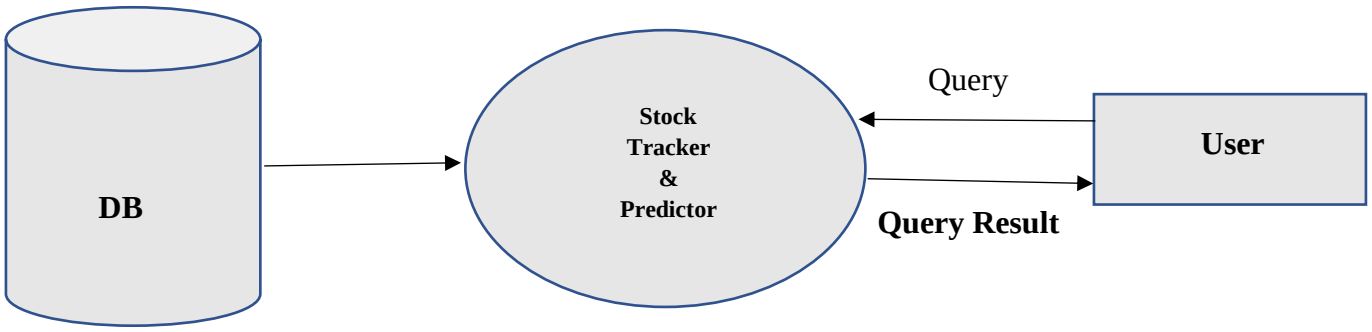


Fig 4.3 Zero Level DFD

4.3.2 LEVEL 1:

A level 1 DFD notates each of the main sub-processes that together form the complete system. We can think of a level 1 DFD as an "exploded view" of the context diagram.

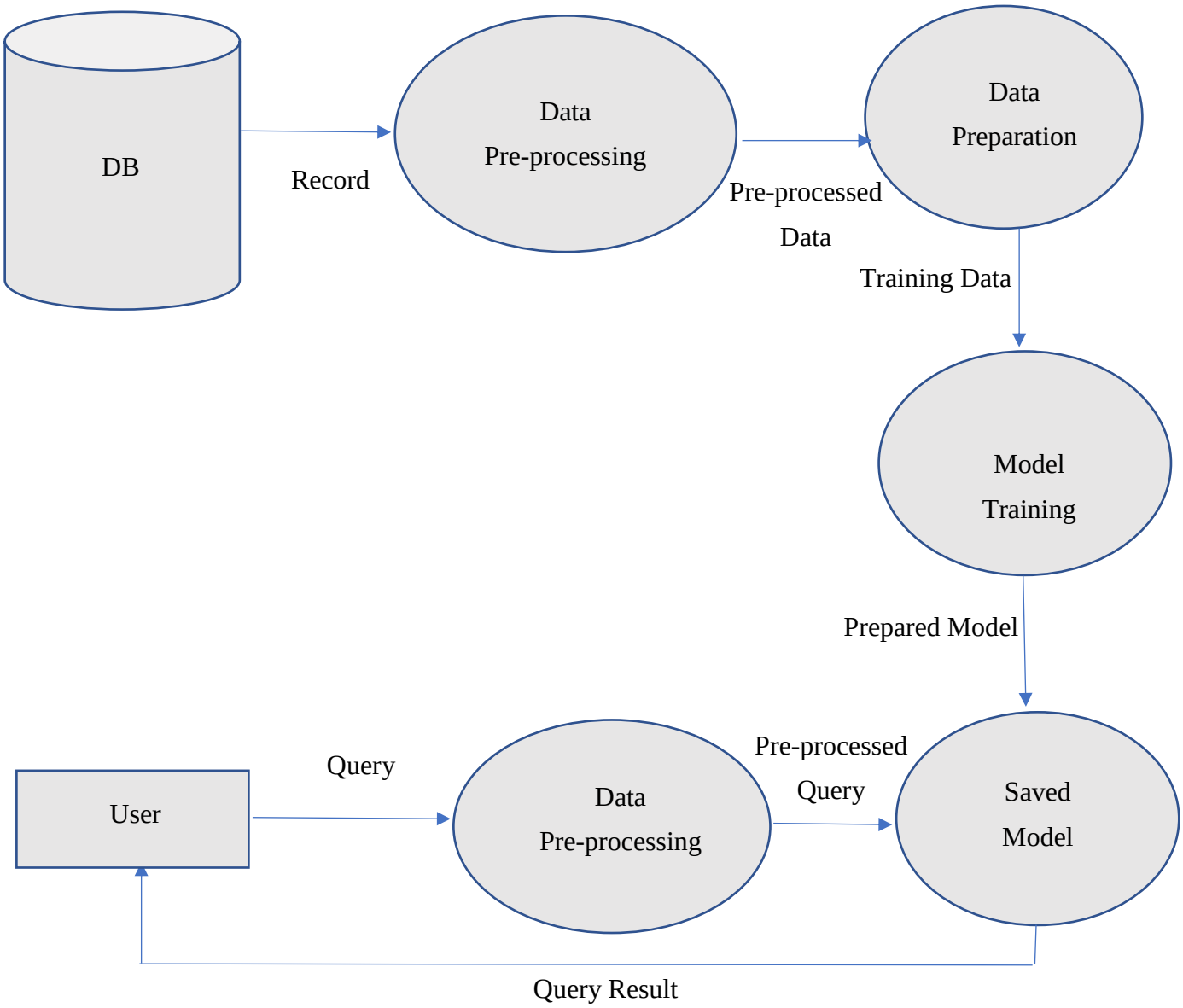


Fig 4.4 First Level DFD

Chapter 5

Implementation

The implementation of the Stock Tracker and Predictor website involves several components, including front-end development, back-end development, and machine learning implementation.

5.1 Front-end Development:

The website's front-end development involves React [1], Tailwind CSS [6], and Firebase [5] to create the website's user interface and authentication. These Technologies makes the web app robust, scalable and highly manageable for future changes and feature addition.

5.2 Back-end Development:

The back-end development of the website involves using a server-side language Node.js [2] to develop the website's logic. This includes creating API endpoints for retrieving data from the database, processing user input, and interacting with the machine learning models. The back-end should also handle user authentication and authorization.

5.3 Machine Learning Implementation:

The machine learning implementation involves developing and training a machine-learning model to predict stock prices. Time Series Forecasting Model Long Short-Term Memory (LSTM) neural networks, which can analyze historical stock prices and make predictions based on patterns in the data is used for prediction and analyzing the stock data.

5.3.1 Importing the requirements & Data Preprocessing

stock > Predictor.py > ...

```python
from numpy import array
import numpy
import yfinance as yf
from pandas_datareader import data as pdr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import datetime as dt
from datetime import date
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.callbacks import EarlyStopping
import pandas_datareader as web
from keras.models import load_model
import streamlit as st


st.title('STOCK TREND PREDICTION')
user_input = st.text_input('ENTER STOCK SYMBOL', 'TITAN.NS')

yf.pdr_override()

# ds = pd.read_csv("./data.csv")
ds = pdr.get_data_yahoo(
    user_input, start=dt.datetime(2022, 1, 1), end=date.today())
# print(ds)

ds2 = ds.iloc[::-1]

fig0 = plt.figure(figsize=(12, 6))
st.subheader('Closing Price vs Time chart')
ds = ds[['Adj Close']]
ds.columns = ['Price']
plt.plot(ds)
st.pyplot(fig0)

scaler = MinMaxScaler(feature_range=(0, 1))
df1 = scaler.fit_transform(np.array(ds).reshape(-1, 1))

training_size = int(len(df1)*0.65)
test_size = len(df1)-training_size
train_data, test_data = df1[0:training_size,
                            :], df1[training_size:len(df1), :1]
```

Fig 5.1 Code Snippet for Data Pre-processing

5.3.2 Model & Prediction

```
74
75  model = Sequential()
76  model.add(LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
77  model.add(Dropout(0.1))
78  model.add(LSTM(50, return_sequences=True))
79  model.add(Dropout(0.1))
80  model.add(LSTM(50, return_sequences=True))
81  model.add(Dropout(0.1))
82  model.add(LSTM(50))
83  model.add(Dropout(0.1))
84  model.add(Dense(1))
85  model.compile(loss='mean_squared_error', optimizer='adam')
86
87  model.fit(X_train, y_train, validation_data=(
88      X_test, ytest), epochs=25, batch_size=32, verbose=1)
89
90  train_predict = model.predict(X_train)
91  test_predict = model.predict(X_test)
92
93  train_predict = scaler.inverse_transform(train_predict)
94  test_predict = scaler.inverse_transform(test_predict)
95
96  # testing part
97  look_back = 10
98  trainPredictPlot = numpy.empty_like(df1)
99  trainPredictPlot[:, :] = np.nan
100 trainPredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
101 # shift test predictions for plotting
102 testPredictPlot = numpy.empty_like(df1)
103 testPredictPlot[:, :] = numpy.nan
104 testPredictPlot[len(train_predict)+(look_back*2) +
105                 1:len(df1)-1, :] = test_predict
106
107 # plot baseline and predictions
108
```

Fig 5.2 Code Snippet for Model

5.4 Testing and Deployment:

Once the website has been developed, it should be thoroughly tested to ensure that it functions as expected and is free from bugs and errors. After testing, the website can be deployed to a web server or cloud platform, such as Amazon Web Services or Microsoft Azure, to make it accessible to users.

Chapter 6

Software and System Requirements

The technical requirements for the proposed project are feasible and not very hard to obtain, although preferably the system should be equipped with a high-speed internet connection. The hardware requirement too is minimal.

6.1 Software Requirements

The technologies required are-

1. React [1]
2. Vite [4]
3. Tailwind CSS [6]
4. Node.js [2]
5. Firebase [5]
6. Python 3.10 [6]
    - 5.1 numpy
    - 5.2 pandas
    - 5.3 matplotlib
    - 5.4 sklearn
    - 5.5 streamlit [9]
    - 5.6 pip
    - 5.7 tensorflow [8]

6.2 System Requirements

The system should meet the following requirements:-

1. Operating System: Windows 7 or above
2. Web Browser: Chrome (preferred)
3. Processor: Intel Core or AMD or M1 chips
4. RAM: 2 GB RAM or above

Chapter 7

Screenshots

7.1 Sign In & Sign Up

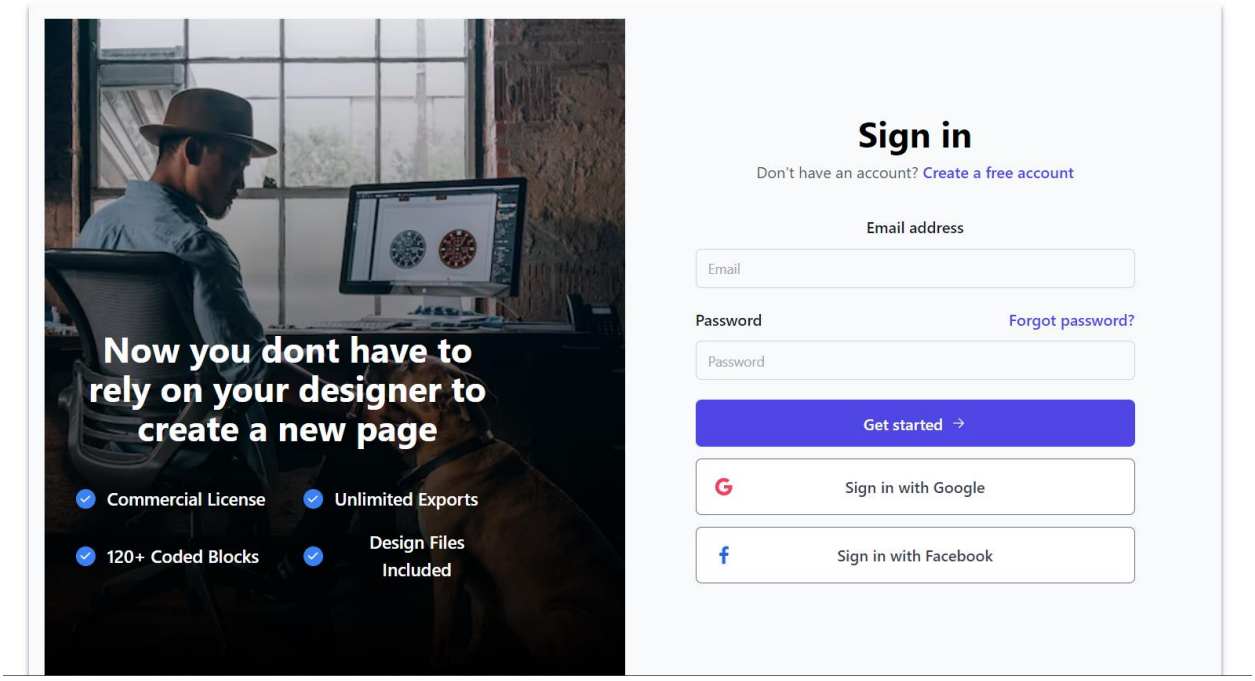


Fig 7.1 Login Screen

7.2 News Page

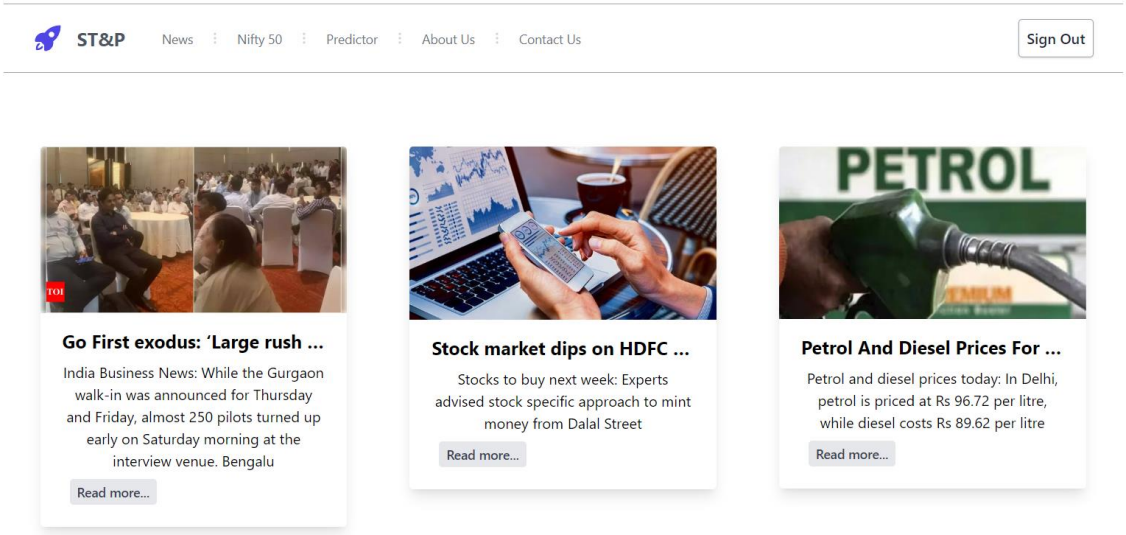
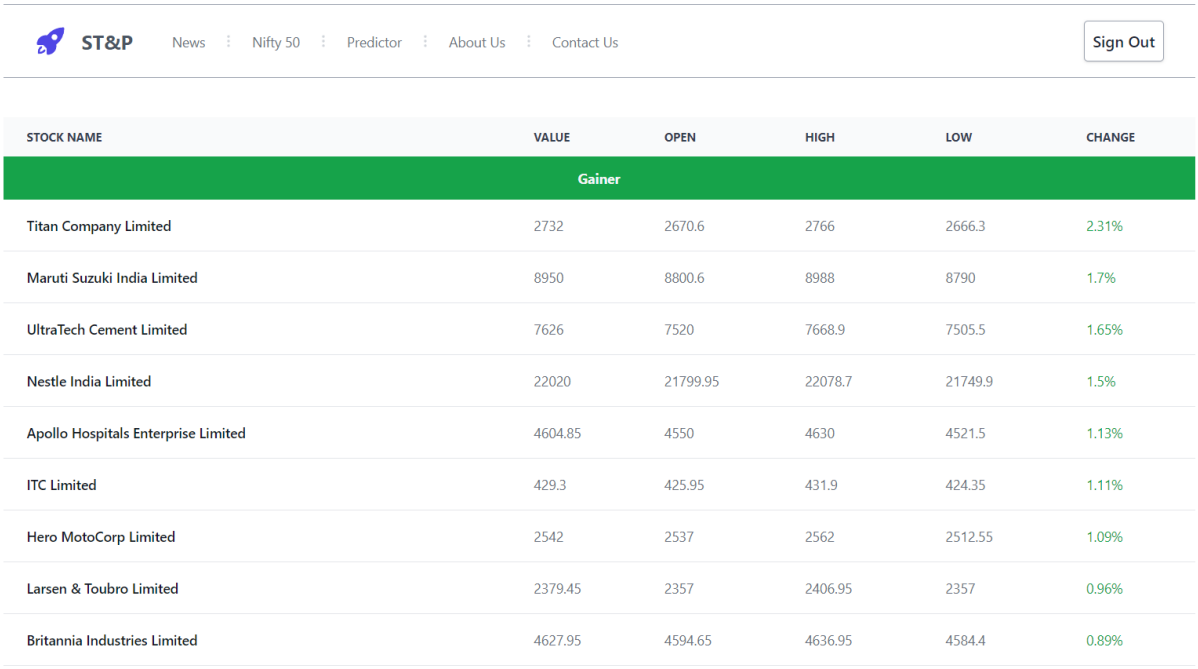


Fig 7.2 News Page

7.3 Nifty 50 Index



STOCK NAME	VALUE	OPEN	HIGH	LOW	CHANGE
Gainer					
Titan Company Limited	2732	2670.6	2766	2666.3	2.31%
Maruti Suzuki India Limited	8950	8800.6	8988	8790	1.7%
UltraTech Cement Limited	7626	7520	7668.9	7505.5	1.65%
Nestle India Limited	22020	21799.95	22078.7	21749.9	1.5%
Apollo Hospitals Enterprise Limited	4604.85	4550	4630	4521.5	1.13%
ITC Limited	429.3	425.95	431.9	424.35	1.11%
Hero MotoCorp Limited	2542	2537	2562	2512.55	1.09%
Larsen & Toubro Limited	2379.45	2357	2406.95	2357	0.96%
Britannia Industries Limited	4627.95	4594.65	4636.95	4584.4	0.89%

Fig 7.3 Nifty 50 Index View

7.4 Chart & Notification

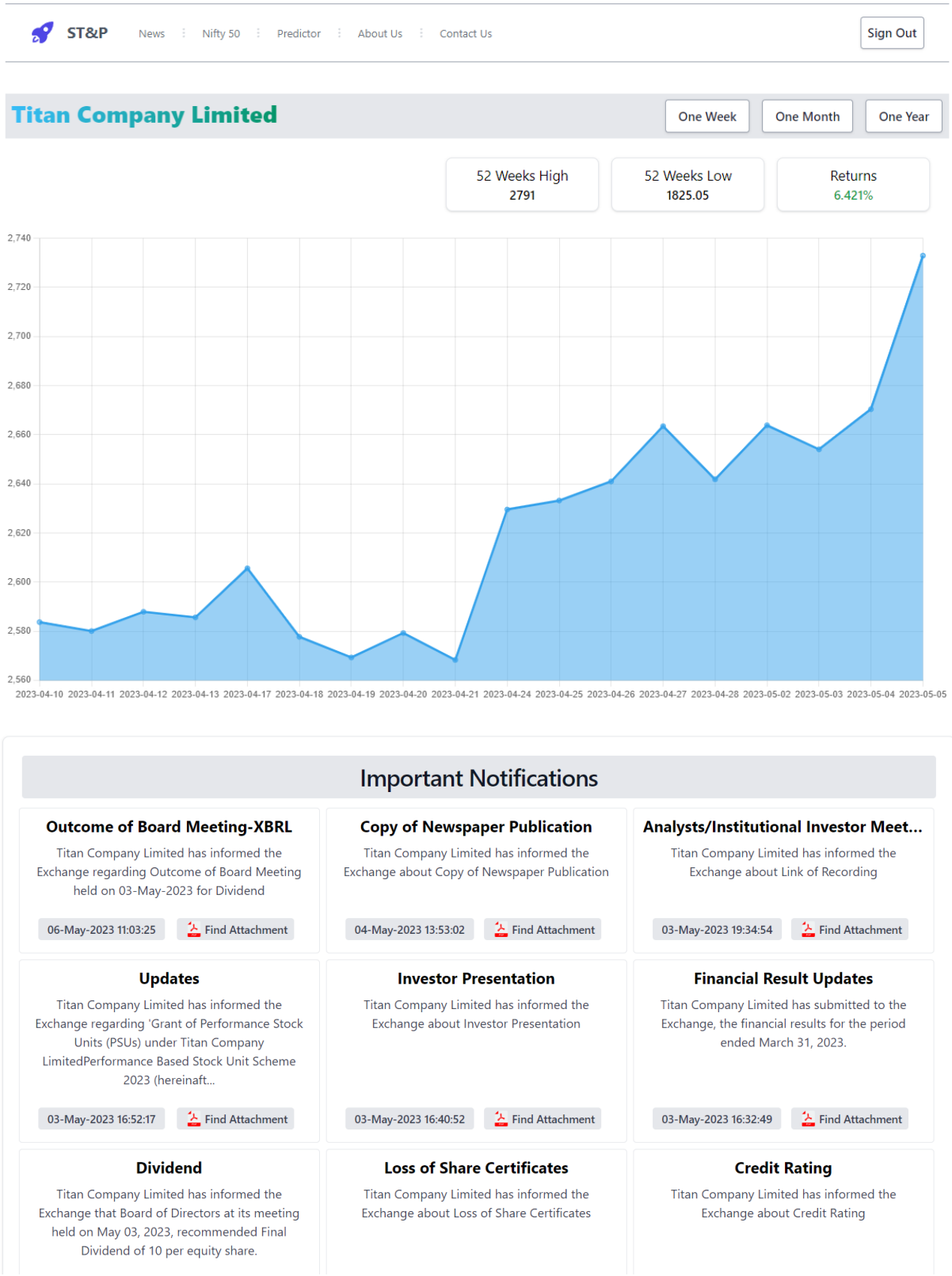


Fig 7.4 Stock Chart & Notification

7.5 Prediction

Fig 7.5 Prediction on Data set

Predicted stock price for day 1 => 572.7789686216402

Predicted stock price for day 2 => 576.3309136022763

Predicted stock price for day 3 => 578.8243771708385

Predicted stock price for day 4 => 580.6189258337308

Predicted stock price for day 5 => 582.5228834046338

Predicted stock price for day 6 => 582.6688611131049

Predicted stock price for day 7 => 583.3919416340977

Predicted stock price for day 8 => 584.8578516590605

Predicted stock price for day 9 => 585.2185509286828

Predicted stock price for day 10 => 586.2614597070187

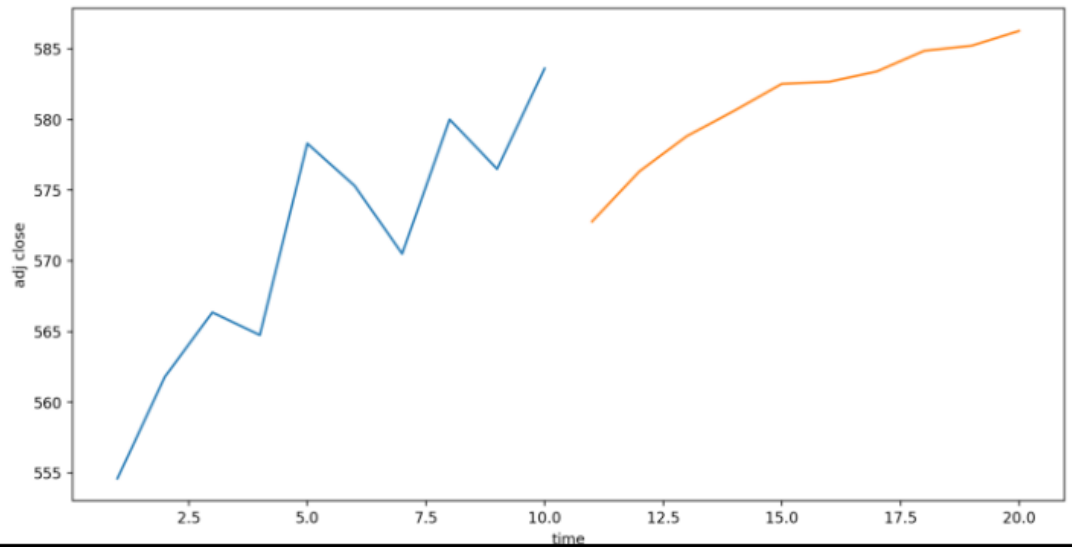


Fig 7.6 Future Prediction

Chapter 8

Conclusion and Future Work

8.1 Conclusion

The web app provides a valuable tool for users to access and analyze stock market data. The app's user-friendly interface and efficient algorithms allow users to easily search for and analyze historical stock data, and generate predictions for future stock prices.

However, it is important to keep in mind that the stock market is unpredictable, and any predictions generated by the app just for educational and informational purposes only, and not as financial advice. Users should conduct thorough research and consult with a qualified financial advisor before making any investment decisions.

Overall, the web app provides a useful resource for individuals interested in the stock market, and its features and functionality make it a valuable tool for those looking to analyze and predict stock prices.

8.2 Future Work

Future scope for the Stock Tracker and Predictor website includes advanced prediction algorithms, integration with real-time data sources, sentiment analysis and news aggregation, user personalization and recommendation features, social engagement capabilities, and expanding coverage to international markets and a wider range of financial instruments.

References

[1] “React” [Online]. Available: https://reactjs.org/

[2] “Node.js” [Online]. Available: https://nodejs.org/

[3] “Express.js” [Online]. Available: https://expressjs.com/

[4] “Vite” [Online]. Available: https://vitejs.dev/

[5] “Firebase” [Online]. Available: https://firebase.google.com/docs/

[6] “Tailwind CSS” [Online]. Available: - https://tailwindcss.com/docs/

[7] “Python” [Online]. Available: https://www.python.org/

[8] “TensorFlow” [Online]. Available: https://www.tensorflow.org/

[9] “Streamlit” [Online]. Available: https://streamlit.io/

[10] “Stack Overflow” [Online]. Available: https://stackoverflow.com/

[11] “GitHub” [Online]. Available: https://github.com/