**CHAPTER 7**

**IMPLEMENTATION**

**7.1 User Interface**

User Interface is a key factor that attracts new users and helps in retaining the existing users. Keeping this in mind, the entire UI/UX of the web-application has been designed using Bootstrap, a popular framework used to design clean and elegant UI for web-apps. To render the charts showing the stock price prediction and buy-sell signals in an interactive way to the user, Chart js., a JavaScript library that is specially made to render charts for stock data has been implemented to enhance the UI of our web-app.

**7.2 Authentication**

Keeping our users safe and secure is our web-app’s top priority. To ensure this, our user’s data is stored securely using Google’s Firebase which handles the web-app’s entire authentication and registration process. Whenever a user logs in to our web-app, his credentials are securely transported to the Firebase’s auth server which verifies the credentials and authenticates the user into our web-app. Similar process is done for the signup of new users also.

**7.3 Stock Price Prediction**

Stock price prediction is a key feature in the web-application. It is implemented using various libraries such as tensorflow, keras, numpy and pandas. Stock price data is collected and fed into our model from Yahoo Finance API which provides stock data for all the stocks in exchanges all over the world. The steps to predict the stock prices are as follows.

The stock data CSV file for various companies are fetched from Yahoo Finance API. From the received dataset, we collect the following data:

1. The date of each company stock price.
2. The opening price of the stock of that particular company.
3. High-KPI: The High-key performance indicator denotes the highest intra-day price reached by the stock of that company.
4. Low-KPI: The Low-key performance indicator denotes the lowest intra-day price reached by the stock.
5. Volume-KPI: The Volume key performance metric represents the number of securities or contracts acquired and sold in the market on that day.
6. Open: The Open Interest-key performance indicator denotes the count of futures shares that are currently outshining in the stock market.

The fetched dataset was later transformed into a NumPy array that will be suitable for using it with our prediction model by performing the following steps:

1. MinMax Scalar is used to set the (feature\_range= (0,1))
2. Reshape the data to the [-1, +1] range.
3. df=scaler.fit\_transform (np. array(df). reshape (-1,1))
4. Transformation of time-series dataset into input and output components for self-supervised learning, where 65% of data is used to train the model and the rest 35% is used for testing the model.
5. Setting the time\_step to 100.
6. Based on the time\_step parameter, we have created the X\_train,y\_train and X\_test and y\_test datasets.

The input dataset is initially split into training and testing datasets. The LSTM and CNN model will then be further used to fit on the training dataset. In this way, the accuracy is evaluated over the test dataset. The CNN and LSTM networks are comprised of three input layers with fifty neurons and an output layer. After the model has been fitted, the tuning of hyper parameters will be begun. We may change back to the original form through the scaler.inverse transformation (rescaling).

**7.4 Algorithmic Trading – Buy and Sell Signals**

When an investor decides to buy or sell a particular stock, he/she must get the best price. To achieve that, our web-application provides buy and sell signals using Moving Averages for all stocks in the Indian Stock Exchange. We construct a simple moving average window for 10 days and 60 days. When the 10-day moving average crosses the 60-day moving average, we indicate a buy signal. When the 60-day moving average crosses the 10-day moving average, we indicate a sell signal. This strategy is followed by traders all over the world. However, the moving average window may vary from trader to trader according to their strategy.

**7.5 Candlestick Stock Screener**

Candlestick charts are very vital for every trader. These charts exhibit various patterns using which traders execute trades. The stock screener provides a list of stocks that satisfy the candlestick pattern selected by the user. This is done using a python library called talib which helps in identifying the candlestick patterns in the stocks. The list of stocks is downloaded from the NSE website, stored as a CSV file. Then, Yahoo Finance API is used to fetch the KPIs and use it to plot the chart patterns. Finally, talib library helps to filter the list of stocks that satisfy the candlestick pattern selected by the user.

**7.6 News**

Stock news are very important for everyone into the market. To satisfy this requirement, stock news is provided within our web-app to keep our users well informed about the latest news in the market. BBC news API has been used to fetch the latest stock news and show them to the user.

**7.7 Overall Integration**

With the various modules hosted in different places, it’s very important to properly integrate each other. The overall integration of the stock price prediction, algo-trading and other modules into the web-app has been done using a popular architecture known as micro-service architecture. This makes all the modules independent, i.e., if there’s any outrage in any of the modules, the entire web-app will not be under maintenance, it can still continue to serve the users and we can easily isolate the defective module, maintain it and get it up and running again.