SPL-2 Project Report

StockTech

Submitted by

Sadia Tabassum

BSSE 1216

Nazmus Sakib

BSSE 1225

Supervised by

Kishan Kumar Ganguly

Assistant Professor

Institute of Information Technology, University of Dhaka

Institute of Information Technology
University of Dhaka

29-05-2023

Supervisor’s Signature

Table of Contents

Index of Figures

1. Introduction

The stock market is an open marketplace where investors can purchase and sell shares of companies that are publicly traded. The companies sell these shares to collect capital for their business and share a percentage of their revenue with those investors. This market has given investors the chance of gaining money and having a prosperous life through investing small initial amounts of money, low risk compared to the risk of opening a new business or the need for a high salary career. Stock markets are affected by many factors causing the uncertainty and high volatility in the market. The investors have to contact the brokerage firms and then make purchase or sale orders. Also they have to rely on unformatted sources to analyze and make decisions. Also the brokerage firms have to manage their clients' portfolios manually.

StockTech is a software to get rid of all the hassle of this stock exchange and bring it together in one platform. StockTech revolutionizes the stock market by providing investors with a comprehensive platform that simplifies the investment process, reduces reliance on brokers, enhances security, and promotes informed decision-making. It aims to empower individuals to take control of their financial future and navigate the stock market with confidence and ease.

1.1. Motivation

The motivation behind StockTech is rooted in the recognition that the current stock market operations can be overwhelming and inaccessible to regular individuals. The complexities of market forecasting and the reliance on brokers for guidance often leave investors feeling disconnected and uncertain about their investment decisions. Additionally, the significant commissions charged by brokers and their potential conflicts of interest raise concerns about the fairness and transparency of the investment process.

StockTech provides a user-friendly platform that consolidates market data, analysis tools, and personalized recommendations. It enables investors to make informed decisions regarding the buying and selling of stocks. The platform enhances investors' understanding of market trends, thereby reducing reliance on brokers and promoting more autonomous and informed decision-making.

1.2. Overview

StockTech is a software platform designed to revolutionize the stock market investing experience. It addresses the challenges faced by investors in navigating the complexities of the stock market and reduces their reliance on traditional methods, such as brokers and manual processes. By consolidating critical functionalities into a comprehensive platform, StockTech empowers investors to make informed decisions, manage their portfolios efficiently, and optimize their returns.

The platform offers a range of features and tools that simplify the investment process. Real-time market data, advanced analytics, and investment recommendations enable users to stay up-to-date with market trends and make well-informed decisions. One of the key advantages of StockTech is its automated transaction system, powered by distributed ledger technology. This eliminates the need for physical presence at stock exchange offices or brokerage firms, streamlining the transaction process and enhancing security. By providing access to comprehensive market information, advanced tools, and automated transaction capabilities, StockTech enables users to manage their finances smartly, enhancing their potential for financial success.

2. Background of the Project

The idea for StockTech emerged from the recognition of the complexities and challenges that individuals face when navigating the stock market. Traditional methods of investing often involve reliance on brokers, who charge high commissions and may not always act in the best interest of investors. Additionally, the need for physical presence at stock exchange offices introduces inefficiencies and potential security vulnerabilities.

2.1. Stock Market Analysis

Stock market analysis involves studying and evaluating various factors to make informed decisions about buying, selling, or holding stocks. Here are three primary methods of stock market analysis:

2.1.1 Fundamental Analysis

This approach involves assessing a company's financial health, performance, and overall value. Fundamental analysts analyze financial statements, such as balance sheets, income statements, and cash flow statements, to determine the intrinsic value of a stock. They also consider macroeconomic factors, industry trends, competitive advantages, and management quality.

2.1.2 Technical Analysis

Technical analysts study stock price charts and patterns to identify trends and predict future price movements. They use various tools and indicators, such as moving averages, support and resistance levels, and volume analysis. Technical analysis assumes that historical price and volume data can provide insights into future market behavior.

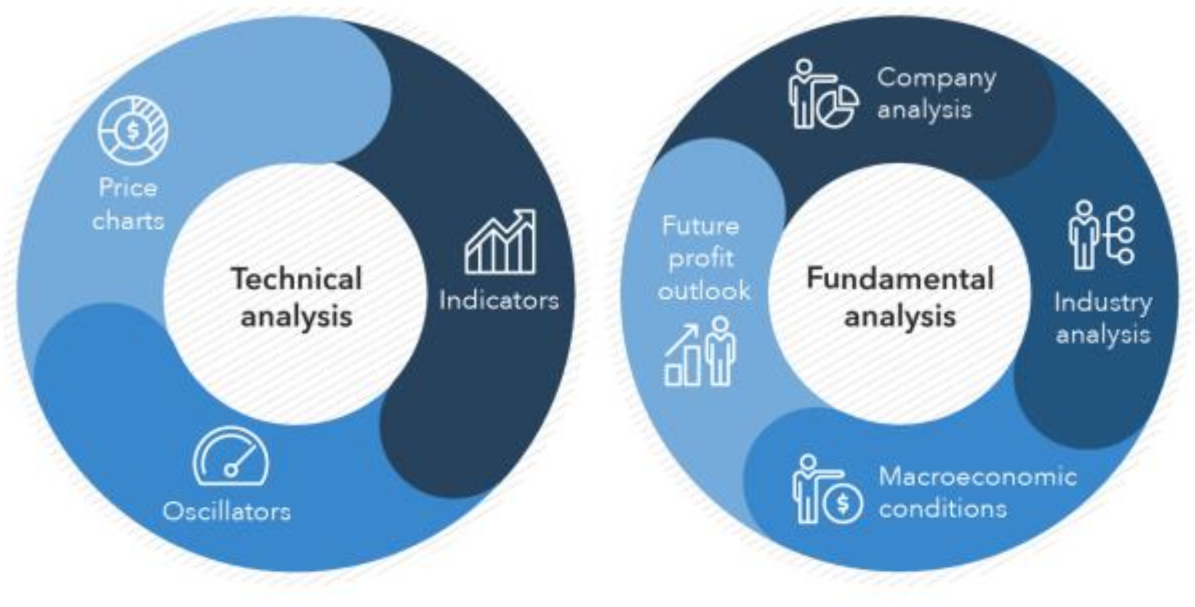


Fig-1: Technical analysis vs Fundamental analysis

There are many technical indicators for stock market analysis. Some commonly used stock market indicators are-

- **Moving Average Convergence Divergence (MACD)**:
  MACD is a trend-following momentum indicator that calculates the difference between two exponential moving averages (usually 12-day and 26-day). It also includes a signal line (typically a 9-day EMA) to generate trading signals when the MACD line crosses above or below the signal line. When the MACD line (fast line) crosses above the signal line (slow line), it generates a bullish signal. Conversely, when the MACD line crosses below the signal line, it generates a bearish signal.

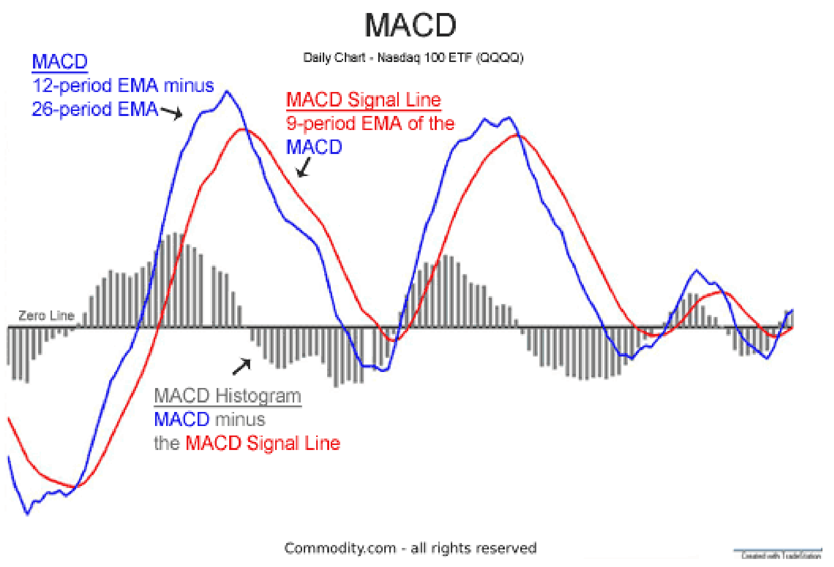


Fig-2: MACD

- **Relative Strength Index (RSI)**:
  RSI measures the speed and change of price movements. It oscillates between 0 and 100 and is used to determine overbought and oversold conditions. Values above 70 typically indicate overbought conditions, while values below 30 suggest oversold conditions.

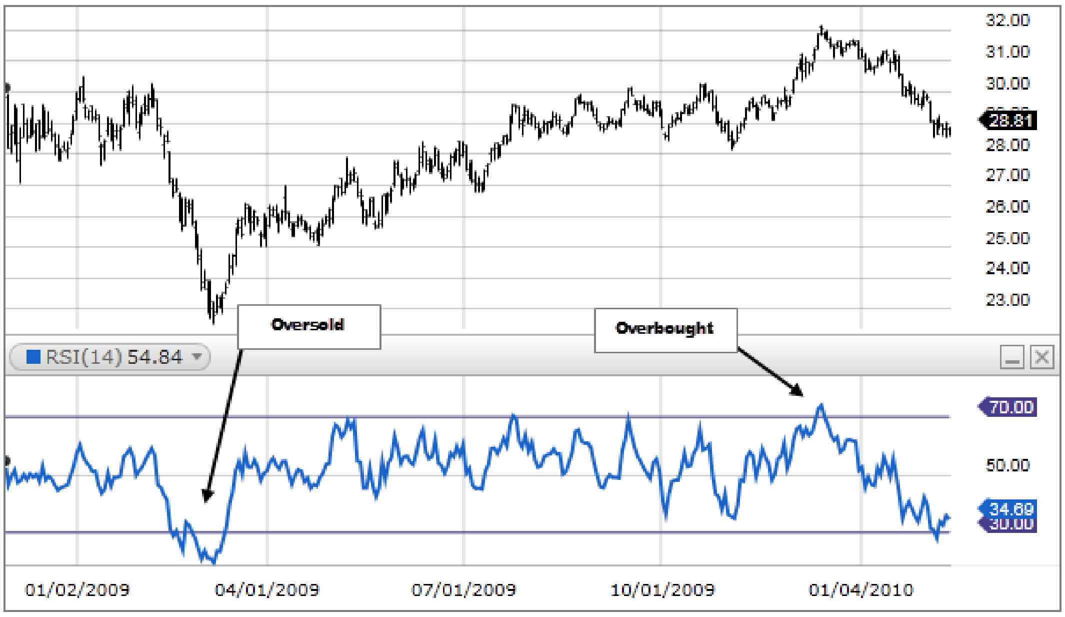


Fig-3: RSI

- **Bollinger Bands (BB)**:

Bollinger Bands consist of a moving average (usually 20-day) with an upper band and a lower band that represent two standard deviations from the moving average. They help identify periods of high or low volatility and potential price reversals. When the price moves near the upper band, it suggests an overbought condition and a potential bearish reversal. Conversely, when the price approaches the lower band, it indicates an oversold condition and a potential bullish reversal.

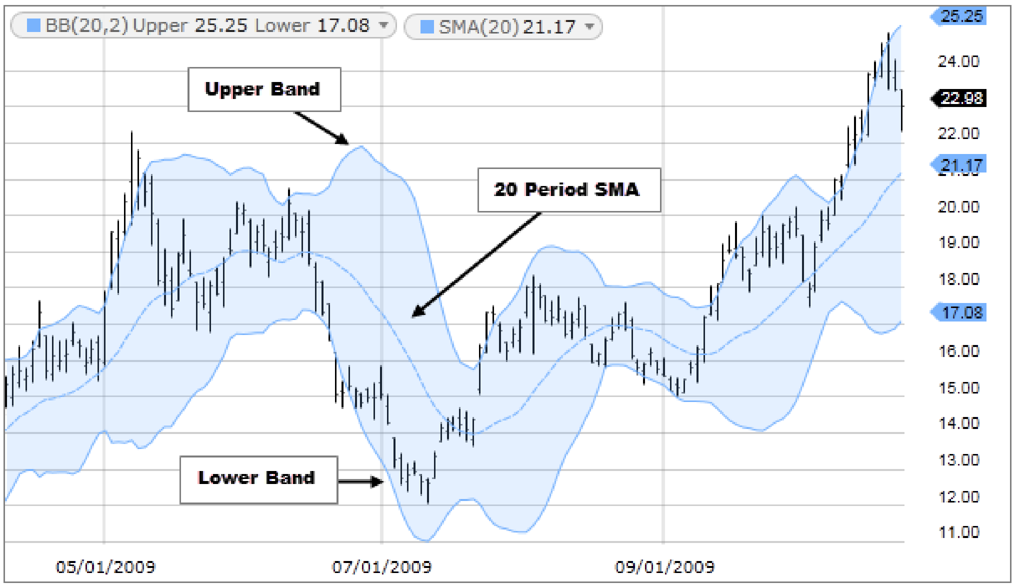


Fig-4: Bollinger Bands

- **Simple Moving Average (SMA)**:
  Simple Moving Average is a basic moving average that calculates the average price over a specific period, giving equal weight to each data point in that period. SMA is widely used to smooth out price fluctuations and identify longer-term trends. Traders often use SMA crossovers, such as the 50-day SMA crossing above or below the 200-day SMA, as signals to enter or exit trades. When the price is consistently above the moving averages, it suggests a bullish trend. Conversely, when the price consistently stays below them, it indicates a bearish trend.

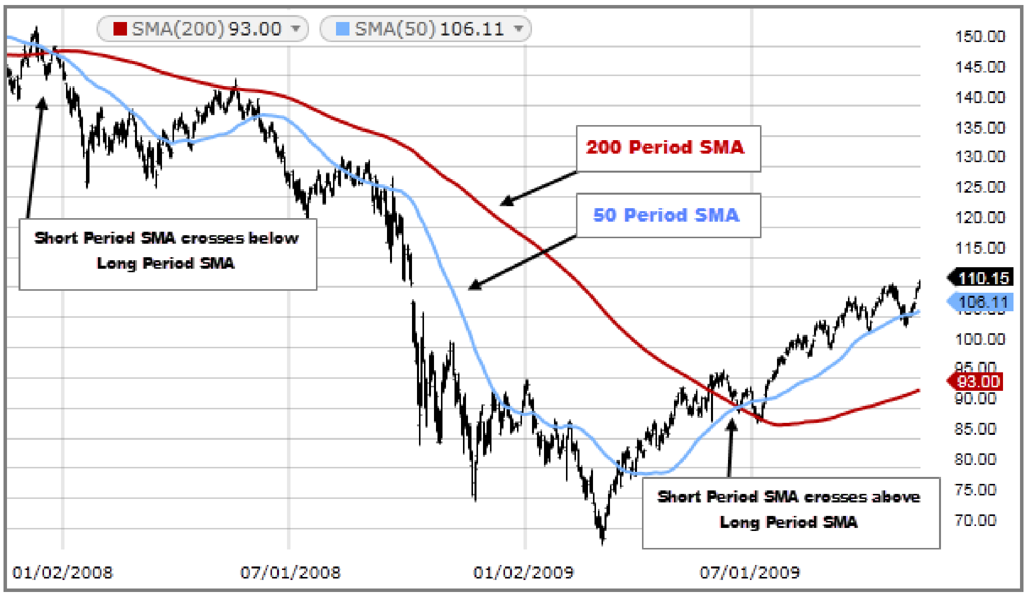


Fig-5: SMA

- **Exponential Moving Average (EMA)**:

Exponential Moving Average is a type of moving average that places more weight on recent price data, making it more responsive to price changes compared to Simple Moving Average (SMA). It calculates the average price over a specified period, with more weight given to recent prices. EMA is often used to identify short-term trends and generate trading signals. Traders commonly use combinations of different EMAs, such as the 9-day EMA and the 50-day EMA, to analyze price movements and identify potential entry and exit points. When the price is consistently above the moving averages, it suggests a bullish trend. Conversely, when the price consistently stays below them, it indicates a bearish trend.

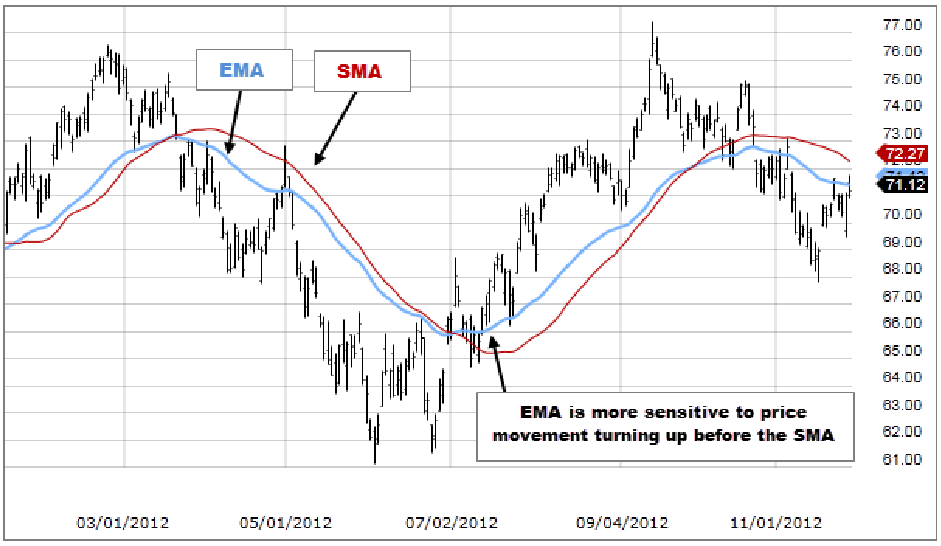


Fig-6: EMA

- **Stochastic Oscillator**:
  The Stochastic Oscillator is a momentum indicator that compares the current closing price of an asset to its price range over a specified period. It consists of two lines: %K and %D. The %K line represents the current price relative to the price range, while the %D line is a moving average of the %K line. It oscillates between 0 and 100. A bullish signal occurs when the %K line crosses above the %D line, suggesting a potential buying opportunity. Conversely, a bearish signal occurs when the %K line crosses below the %D line, indicating a potential selling opportunity.

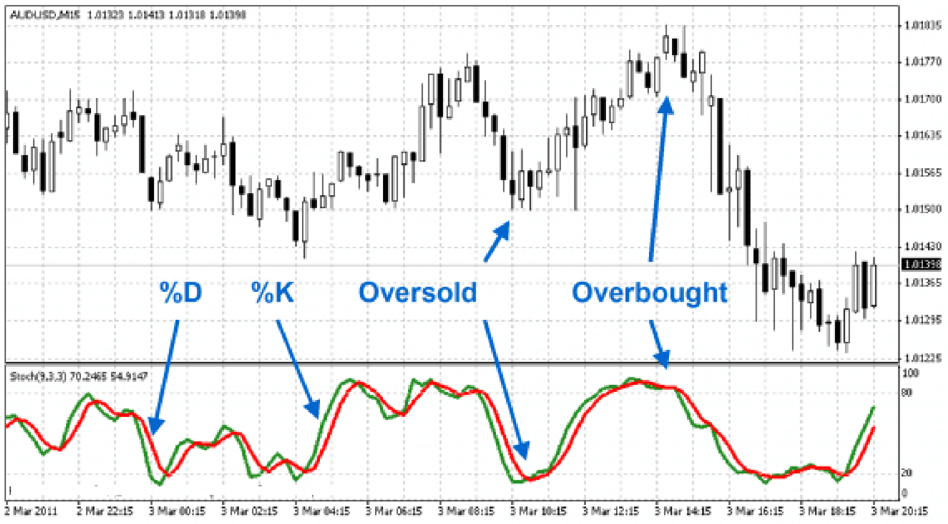


Fig-7: Stochastic Oscillator

2.1.3 Sentiment Analysis

This type of analysis focuses on investor sentiment and market psychology. Sentiment analysts look at factors such as news sentiment, social media trends, and market indicators to gauge the overall mood of investors. Positive sentiment may suggest an optimistic market, while negative sentiment could indicate a bearish or cautious market.

It's important to note that no single analysis method guarantees accurate predictions. Moreover, external factors like geopolitical events, economic conditions, and regulatory changes can significantly impact stock markets. We used technical analysis in this project.

2.2. Utilizing Blockchain Technology

We observed the increasing interest in stock market investing among individuals from various backgrounds, driven by the desire for financial growth and security. However, we also recognized the barriers that hindered individuals from fully participating in the market. These barriers included limited access to comprehensive market information, a lack of transparent and secure transaction processes, and the overwhelming nature of market forecasting.The project incorporates the utilization of the blockchain for transactions.

Blockchain technology provides a highly secure platform for conducting transactions. The decentralized nature of the blockchain network ensures that there is no single point of failure or vulnerability. Transactions recorded on the blockchain are encrypted, time-stamped, and linked to previous transactions, creating an immutable and tamper-proof ledger. This increased security reduces the risk of unauthorized access, fraud, or manipulation of stock market analysis data and transactions.

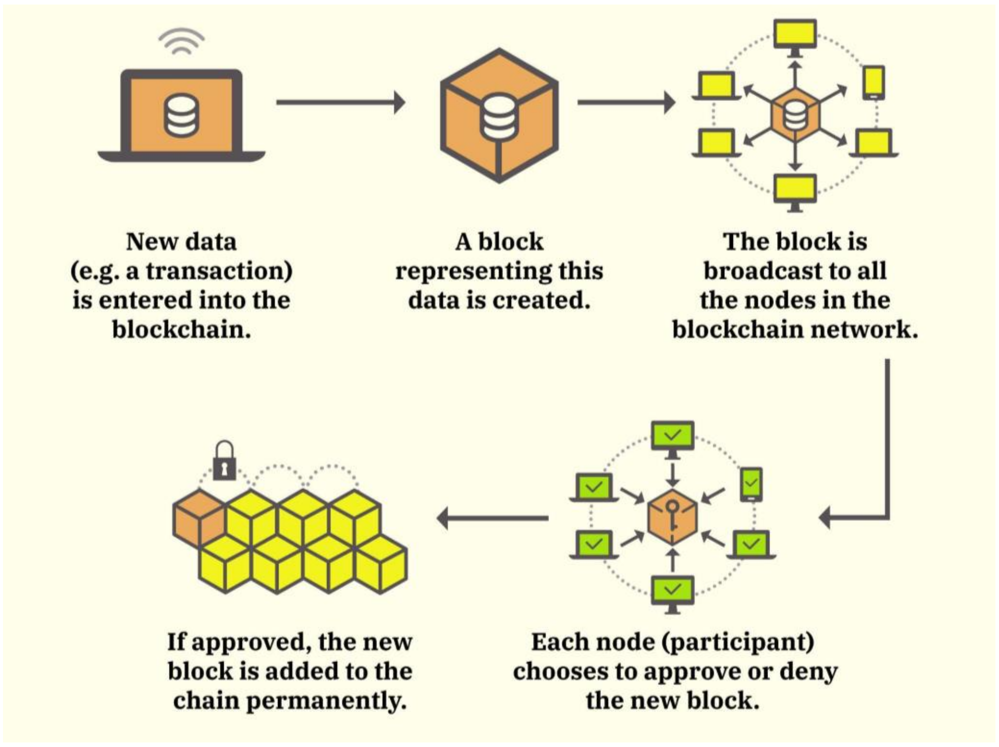

New data (e.g. a transaction) is entered into the blockchain.
A block representing this data is created.
The block is broadcast to all the nodes in the blockchain network.
Each node (participant) chooses to approve or deny the new block.
If approved, the new block is added to the chain permanently.

Fig-8: Blockchain process

By leveraging blockchain, stock market analysis transactions become transparent and verifiable. Every transaction recorded on the blockchain is visible to all participants in the network, eliminating the need for intermediaries and fostering trust among stakeholders. This transparency allows investors and regulators to validate the accuracy and integrity of transactions, thereby enhancing confidence in the stock market analysis process.

2.2.1 Smart Contracts

Smart contracts, which are self-executing agreements stored on the blockchain, play a vital role in streamlining stock market analysis transactions. By incorporating smart contracts, the analysis process becomes automated and eliminates the need for manual intervention or reliance on intermediaries. Smart contracts contain predefined rules and conditions that govern the execution and settlement of transactions. They automatically enforce these terms, ensuring accuracy and efficiency in transaction settlements. This automation reduces human error, accelerates the transaction process, and minimizes the costs associated with intermediaries.

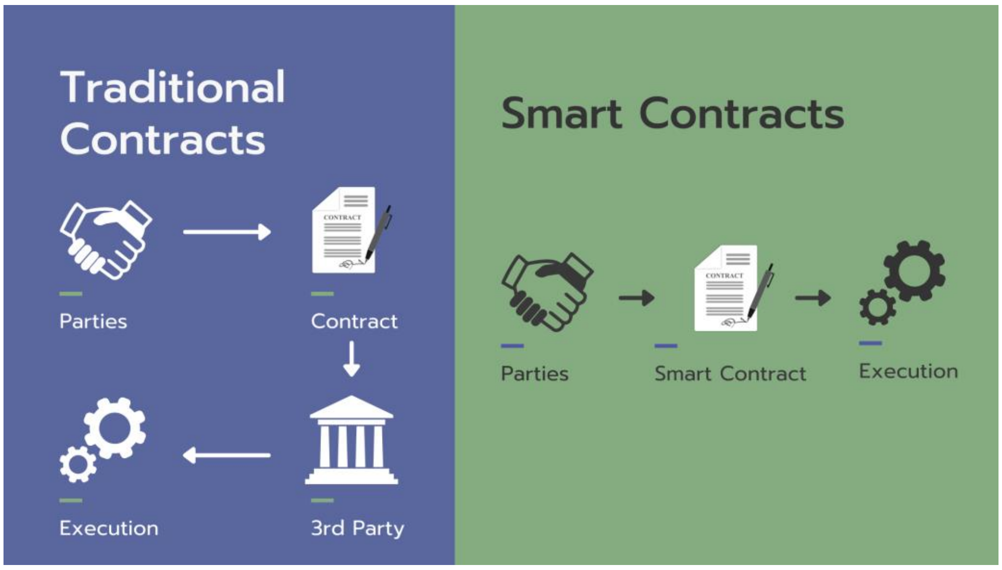


Fig-9: Traditional contract vs smart contract

We created a smart contract specifically designed to facilitate stock market analysis transactions. The smart contract contained predefined rules and conditions governing the execution and settlement of transactions. Instead of deploying our smart contract on the mainnet, which operates on the live Ethereum network, we opted to utilize the Goerli testnet.

2.2.2 Goerli Testnet

The Goerli testnet is a test network that mirrors the functionality of the mainnet but utilizes test Ether (ETH) rather than real Ether. StockTech utilizes the Goerli Testnet that provides a safe and controlled environment for testing and validating the project's automated transaction system. By leveraging the power of blockchain technology through the Goerli Testnet, StockTech ensures secure, transparent, and efficient transactions. This eliminates the need for physical presence at stock exchange offices and minimizes the risk of security breaches, such as check fraud.

2.3. Stock Price prediction

Predicting stock market movements is another complex task, and while LSTM (Long Short-Term Memory) models can be useful for time series analysis, it's important to note that no model can accurately predict stock prices with absolute certainty. However, LSTM models can capture patterns and dependencies in historical stock price data, providing insights into potential future trends.

Long Short-Term Memory (LSTM): StockTech uses the power of LSTM models for price prediction. LSTM, a type of recurrent neural network (RNN), is specifically designed for analyzing sequential data, such as time series data. By training LSTM models on historical stock market data, the project aims to accurately predict future price movements. These predictions serve as the basis for generating investment recommendations and assisting users in making informed decisions regarding their stock portfolios.

LSTM architecture: The LSTM network architecture consists of three parts, as shown in the image below, and each part performs an individual function. The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp. This one cycle of LSTM is considered a single-time step. These three parts of an LSTM unit are known as gates. They control the flow of information in and out of the memory cell or LSTM cell. The first gate is called *Forget gate*, the second gate is known as the *Input gate*, and the last one is the *Output gate*. An LSTM unit that consists of these three gates and a memory cell or LSTM cell can be considered as a layer of neurons in a traditional feedforward neural network, with each neuron having a hidden layer and a current state.

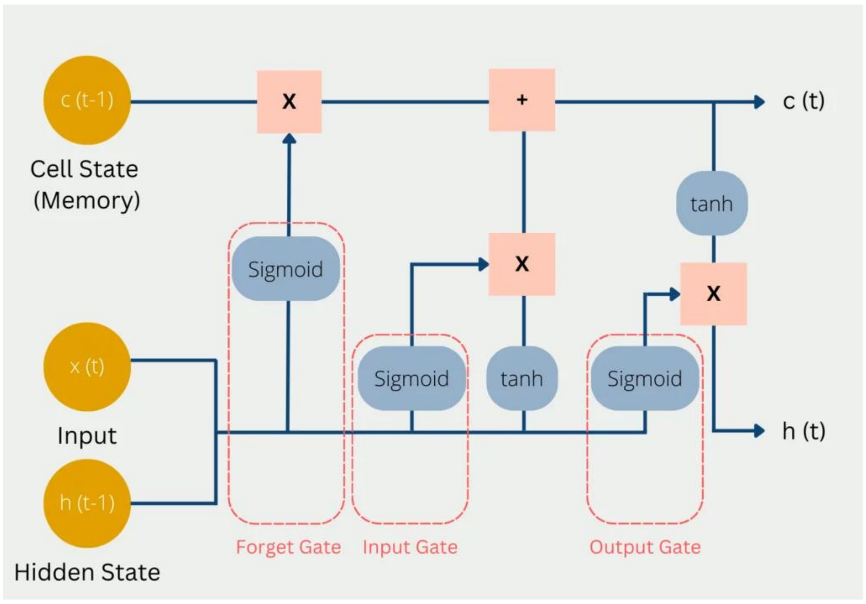


Fig-10: LSTM architecture

Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that, LSTM also has a cell state represented by C(t-1) and C(t) for the previous and current timestamps, respectively. Here the hidden state is known as Short term memory, and the cell state is known as Long term memory.

Forget Gate

In a cell of the LSTM neural network, the first step is to decide whether we should keep the information from the previous time step or forget it. Here is the equation for the forget gate.

$$f_t=\sigma(X_t*U_f+H_{t-1}*W_f)$$

Here,

X_t: input to the current timestamp.

U_f: weight associated with the input

H_{t-1}: The hidden state of the previous timestamp

W_f: It is the weight matrix associated with the hidden state

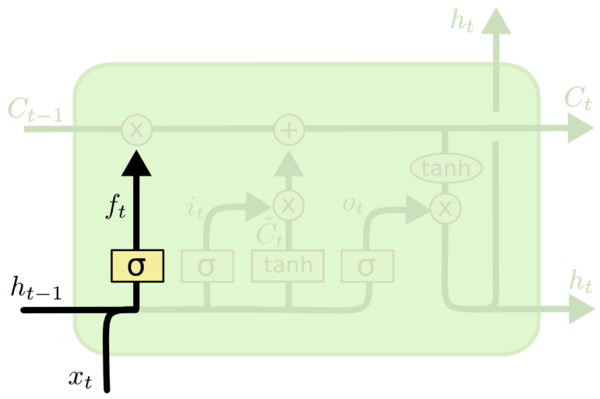


Fig-11: Forget gate of LSTM

Later, a sigmoid function is applied to it. That will make it a number between 0 and 1. This f_t is later multiplied with the cell state of the previous timestamp.

Input Gate

The input gate is used to quantify the importance of the new information carried by the input. Here is the equation of the input gate

$$i_t=\sigma(X_t*U_i+H_{t-1}*W_i)$$

Here,

X_t: Input at the current timestamp t

U_i: weight matrix of input

H_{t-1}: A hidden state at the previous timestamp

W_i: Weight matrix of input associated with hidden state

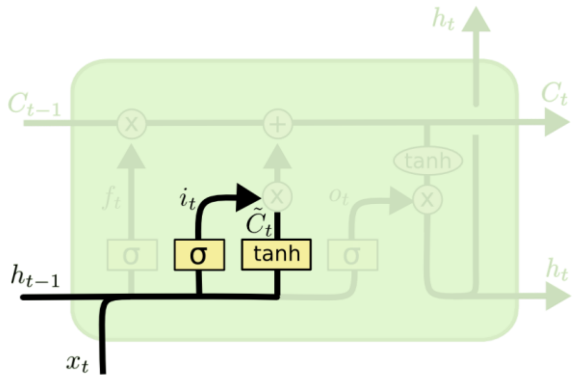


Fig-12: Input gate of LSTM

Again we need to apply the sigmoid function over it. As a result, the value of I at timestamp t will be between 0 and 1.

New Information

Now the new information that needs to be passed to the cell state is a function of a hidden state at the previous timestamp t-1 and input x at timestamp t. The activation function here is tanh. Due to the tanh function, the value of new information will be between -1 and 1. If the value of Nt is negative, the information is subtracted from the cell state, and if the value is positive, the information is added to the cell state at the current timestamp.

$\mathbf{N_t=tanh(x_t*U_c+H_{t-1}*W_c)}$

$\mathbf{C_t=f_t*C_{t-1}+i_t*N_t)}$

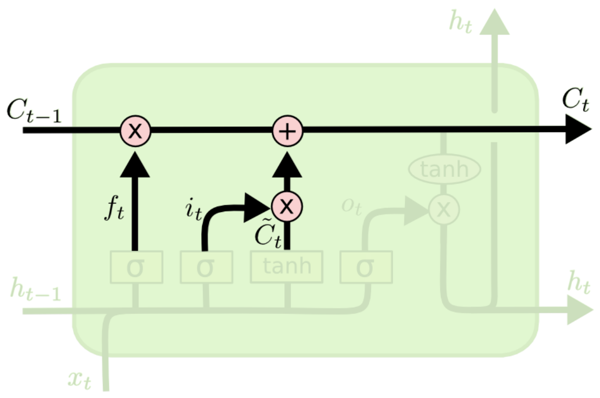


Fig-13: New Information of LSTM

Here, C_{t-1} is the cell state at the current timestamp, and the others are the values we have calculated previously.

Output Gate

Finally, we need to decide what we're going to output. This output will be based on our cell state, but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh (to push the values to be between −1 and 1) and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

Here is the equation for output gate-

$$\mathbf{o_t=\sigma(X_t*U_o+H_{t-1}*W_o)}$$

$$\mathbf{H_t=0_t *tanh(C_t)}$$

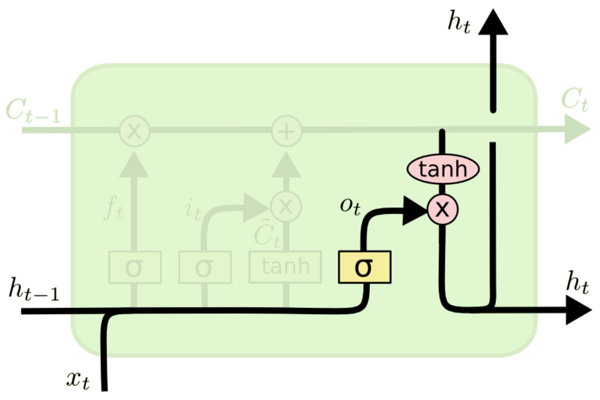


Fig-14: Output gate of LSTM

Its value will also lie between 0 and 1 because of this sigmoid function. Now to calculate the current hidden state, we will use Ot and tanh of the updated cell state. As shown below.

3. Description of the Project

This software is an order management system for a brokerage firm that is already registered in DSE. An order management system (OMS) is a software system that facilitates and manages the execution of trade orders. In the financial markets, an order must be placed in a trading system to execute a buy or sell order for a security. This software gives an interface to the investors and passes all the information to DSE and CDBL using the back office of Flex TP. CDBL gets all the pay in and pay out information and DSE gets the client file and position file daily.The modules of the software are described below.

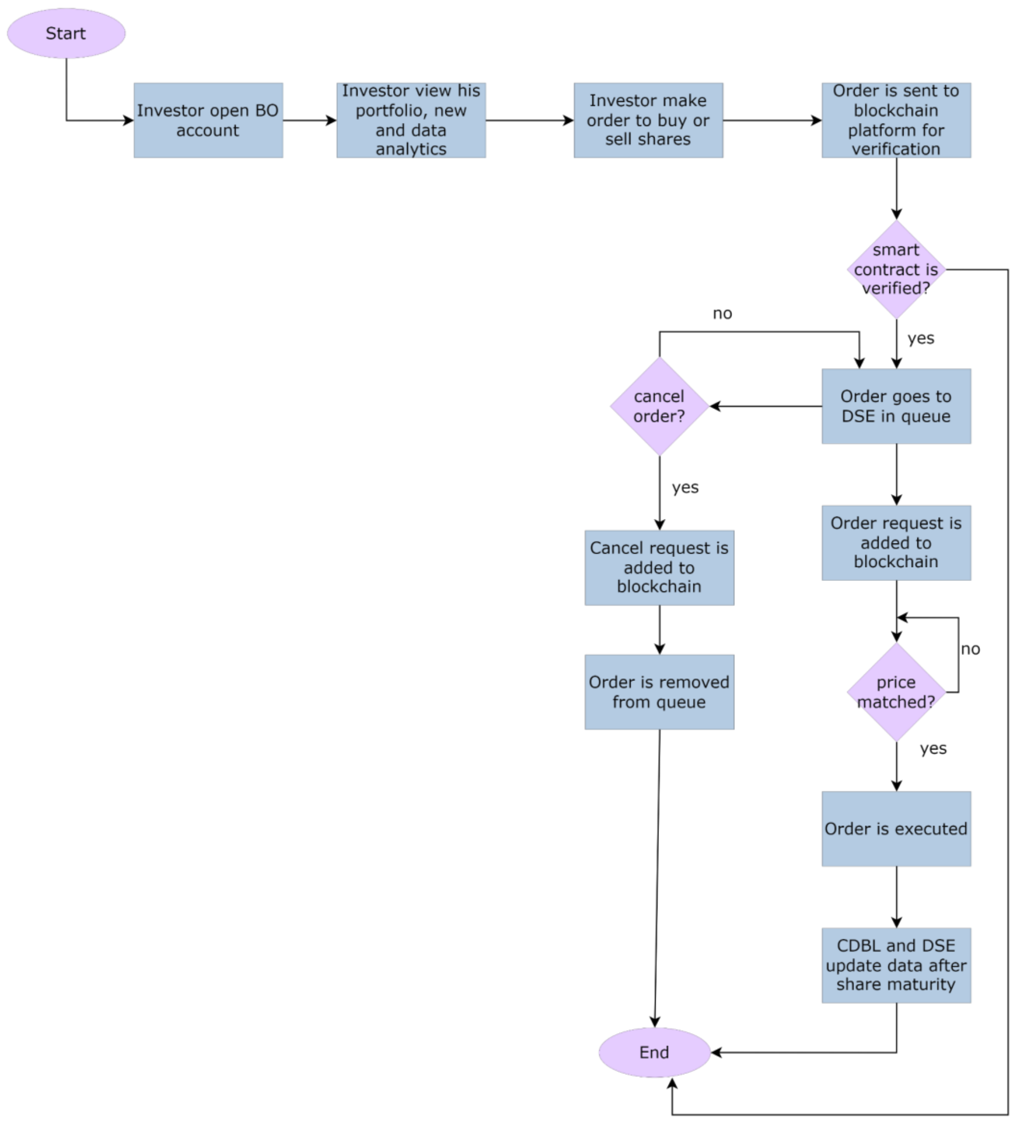
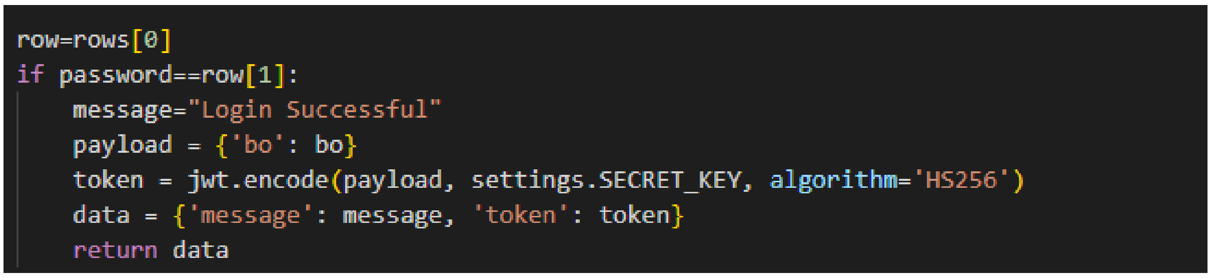

Start
Investor open BO account
Investor view his portfolio, new and data analytics
Investor make order to buy or sell shares
Order is sent to blockchain platform for verification
smart contract is verified?
no
yes
cancel order?
Order goes to DSE in queue
yes
Cancel request is added to blockchain
Order request is added to blockchain
no
Order is removed from queue
price matched?
yes
Order is executed
CDBL and DSE update data after share maturity
End

Fig-15: Proposed system

3.1. Account Management

The StockTech software requires investors to sign in using their BO account number and password. Two-way verification is conducted through an OTP sent to their registered phone number. Password reset can be done via OTP verification. New investors can open a BO account by filling out a form and providing necessary information. After approval, they receive their BO account number via email. JWT token is generated after signing in to the system for authorization.

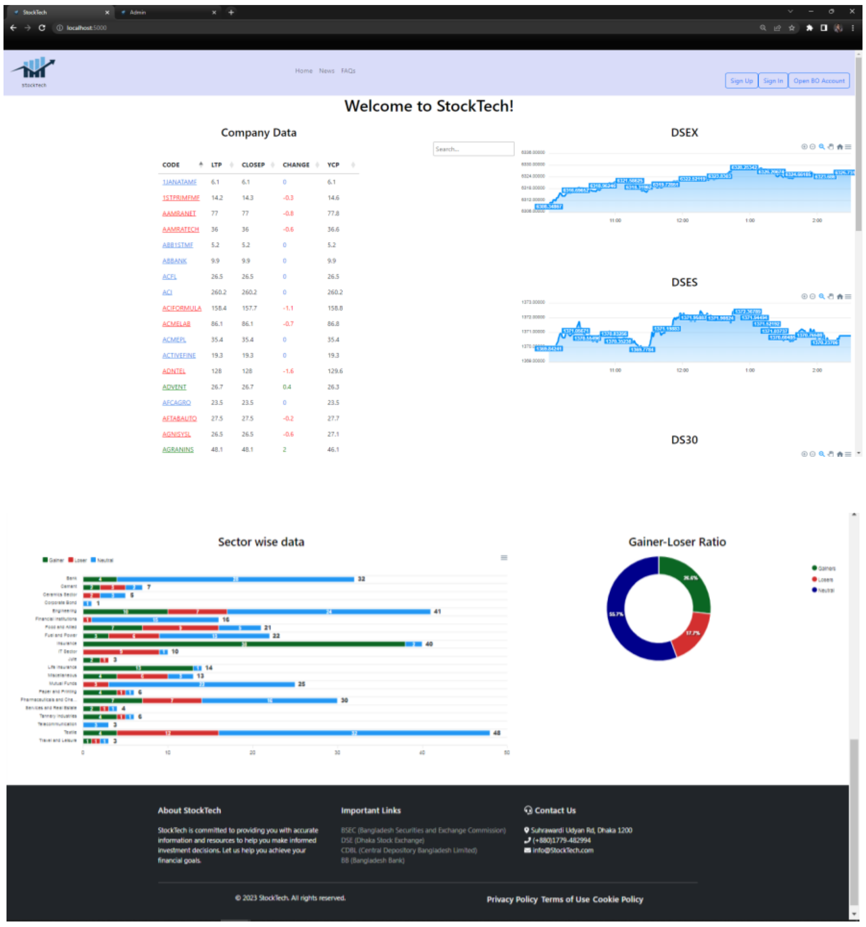


```
row=rows[0]
if password==row[1]:
    message="Login Successful"
    payload = {'bo': bo}
    token = jwt.encode(payload, settings.SECRET_KEY, algorithm='HS256')
    data = {'message': message, 'token': token}
    return data
```

Fig-16: Code snippet- JWT token generation

3.2. Dashboard

The dashboard of the stock market website provides different views and functionalities depending on the user's status. Guest users have access to stock price information and financial news, allowing them to stay updated with the latest market trends. Signed-in users, on the other hand, gain additional features and personalized information. They can view their portfolios, track their order history, stay informed about announcements related to their investments, access stock analysis tools, and execute trades. The dashboard caters to both informative and interactive needs, providing a comprehensive platform for users to manage their investments and make informed trading decisions.



Welcome to StockTech!

Company Data

CODE	LTP	CLOSEP	CHANGE	YCP
1JANATAMF	6.1	6.1	0	6.1
1STPRIMFMF	14.2	14.3	-0.3	14.6
AAMRANET	77	77	-0.8	77.8
AAMRATECH	36	36	-0.6	36.6
ABB1STMF	5.2	5.2	0	5.2
ABBANK	9.9	9.9	0	9.9
ACFL	26.5	26.5	0	26.5
ACI	260.2	260.2	0	260.2
ACIFORMULA	158.4	157.7	-1.1	158.8
ACMELAB	86.1	86.1	-0.7	86.8
ACMEPL	35.4	35.4	0	35.4
ACTIVEFINE	19.3	19.3	0	19.3
ADNTEL	128	128	-1.6	129.6
ADVENT	26.7	26.7	0.4	26.3
AFCAGRO	23.5	23.5	0	23.5
AFTABAUTO	27.5	27.5	-0.2	27.7
AGNISYSL	26.5	26.5	-0.6	27.1
AGRANINS	48.1	48.1	2	46.1

About StockTech

StockTech is committed to providing you with accurate information and resources to help you make informed investment decisions. Let us help you achieve your financial goals.

Important Links

BSEC (Bangladesh Securities and Exchange Commission)
DSE (Dhaka Stock Exchange)
CDBL (Central Depository Bangladesh Limited)
BB (Bangladesh Bank)

Contact Us

Suhrawardi Udyan Rd, Dhaka 1200
(+880)1779-482994
info@StockTech.com

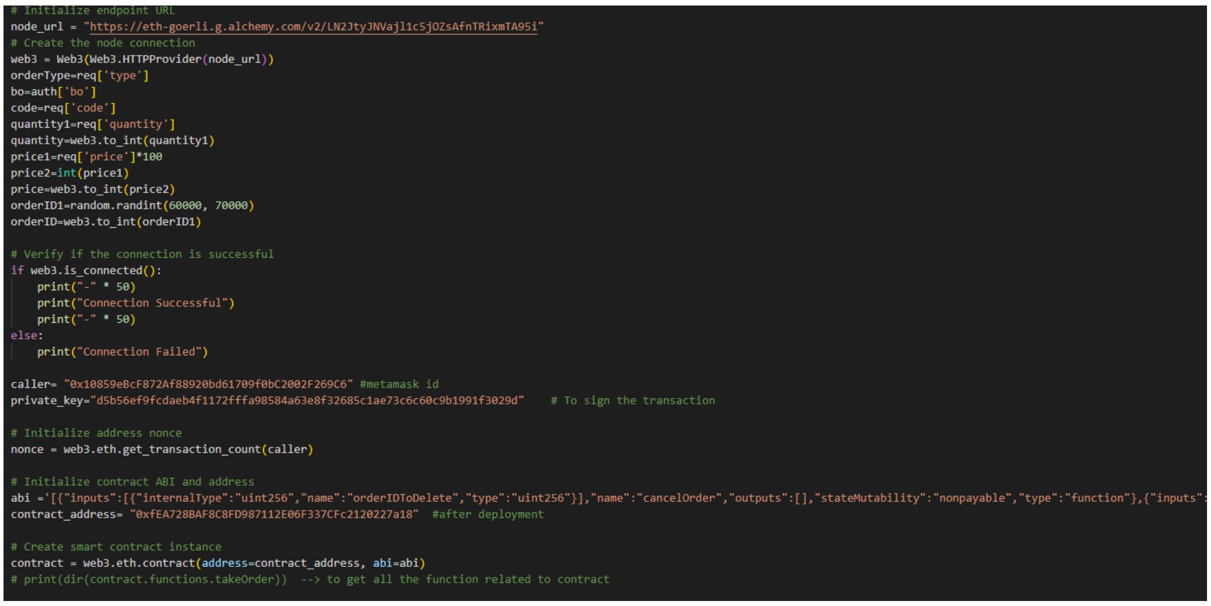
© 2023 StockTech. All rights reserved.

Privacy Policy Terms of Use Cookie Policy

Fig-17: HomePage

3.3. Order Management

Order Management in the stock market website allows investors to place buy/sell orders directly through the platform. Investors can specify the company name, quantity of shares, and the desired price for the transaction. These orders are processed using a smart contract, eliminating the need for traditional brokers. The smart contract ensures transparency and security by being verified on a public blockchain platform. Once the orders are validated, they are sent to the stock exchange (such as DSE - Dhaka Stock Exchange) for execution. Both the buyer and seller receive confirmation of the executed transaction, and their respective accounts are updated accordingly. This process provides investors with a convenient and secure way to manage their orders, reducing the reliance on intermediaries and enhancing efficiency in the trading process.



```python
# Initialize endpoint URL
node_url = "https://eth-goerli.g.alchemy.com/v2/LN2JtyJNVajl1c5jOZsAfnTRixmTA95i"
# Create the node connection
web3 = Web3(Web3.HTTPProvider(node_url))
orderType=req['type']
bo=auth['bo']
code=req['code']
quantity1=req['quantity']
quantity=web3.to_int(quantity1)
price1=req['price']*100
price2=int(price1)
price=web3.to_int(price2)
orderID1=random.randint(60000, 70000)
orderID=web3.to_int(orderID1)

# Verify if the connection is successful
if web3.is_connected():
    print("-" * 50)
    print("Connection Successful")
    print("-" * 50)
else:
    print("Connection Failed")

caller= "0x10859eBcF872Af88920bd61709f0bC2002F269C6" #metamask id
private_key="d5b56ef9fcdaeb4f1172fffa98584a63e8f32685c1ae73c6c60c9b1991f3029d"    # To sign the transaction

# Initialize address nonce
nonce = web3.eth.get_transaction_count(caller)

# Initialize contract ABI and address
abi ='[{"inputs":[{"internalType":"uint256","name":"orderIDToDelete","type":"uint256"}],"name":"cancelOrder","outputs":[],"stateMutability":"nonpayable","type":"function"},{"inputs":
contract_address= "0xfEA728BAF8C8FD987112E06F337CFc2120227a18"  #after deployment

# Create smart contract instance
contract = web3.eth.contract(address=contract_address, abi=abi)
# print(dir(contract.functions.takeOrder))  --> to get all the function related to contract
```

Fig-18: Code snippet- Interact with smart contract

3.4. Analysis

The software provides stock data analysis using algorithms and indicators, aiding investors in decision-making. Technical analysis generates probable suggestions for better market decisions. Stock market indicators are tools and metrics used by investors and analysts to assess the overall health, trends, and potential direction of the stock market. The software incorporates popular indicators such as MACD, RSI, SMA, EMA, Bollinger Bands, and Stochastic Oscillator. MACD helps identify trend reversals, RSI indicates overbought or oversold conditions, SMA and EMA provide trend information, Bollinger Bands indicate price volatility, and Stochastic Oscillator identifies potential reversals. These indicators aid investors in assessing market trends, identifying trading opportunities, and making informed decisions.

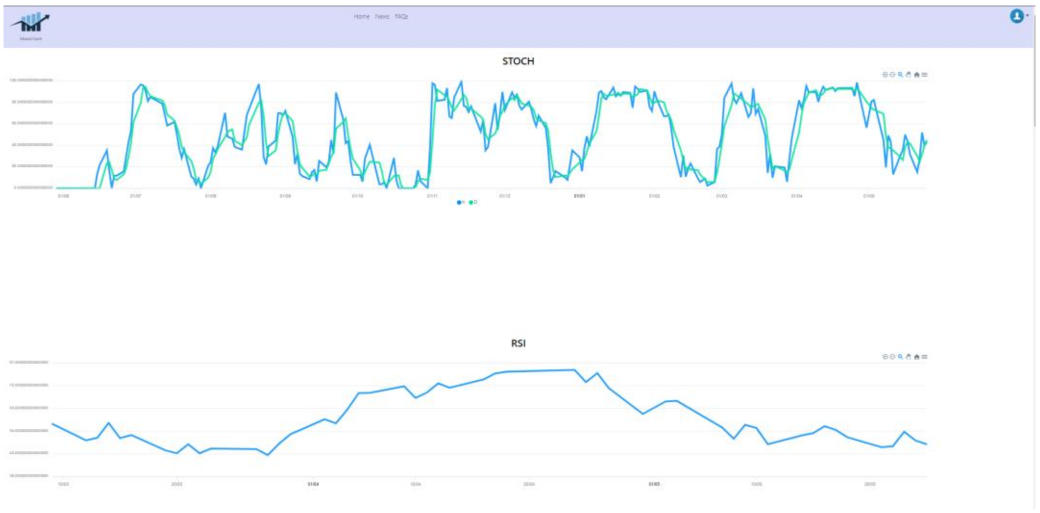
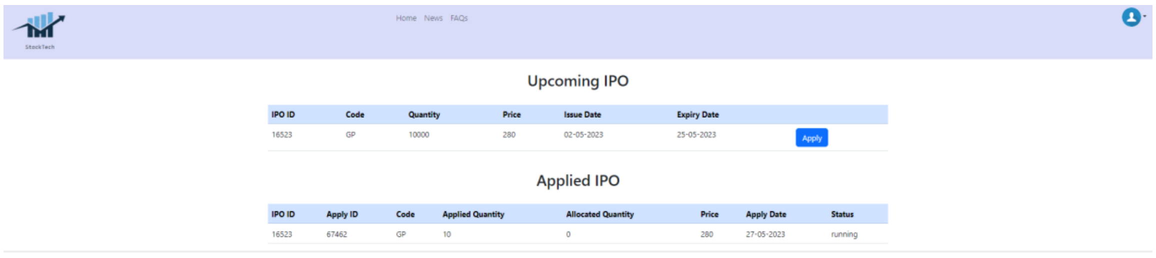


Fig-19: Graph Page

3.5. IPO Management

IPO Management involves the announcement of upcoming IPOs by the administrator, allowing eligible investors to apply through smart contracts. After the IPO period, shares are distributed among brokerage firms by the stock exchange, and applicants receive allocations based on their applied amount and the issued shares ratio. The use of smart contracts ensures transparency, security, and accuracy throughout the process, streamlining applications and automating share allocation rules, ultimately facilitating a fair and efficient distribution of shares during an IPO.



Upcoming IPO

IPO ID	Code	Quantity	Price	Issue Date	Expiry Date	
16523	GP	10000	280	02-05-2023	25-05-2023	Apply

Applied IPO

IPO ID	Apply ID	Code	Applied Quantity	Allocated Quantity	Price	Apply Date	Status
16523	67462	GP	10	0	280	27-05-2023	running

Fig-20: IPO page

3.6. Price Estimation

Stock price prediction refers to the process of forecasting the future value of a particular stock or a collection of stocks traded in the financial markets. It involves utilizing various techniques, including statistical analysis, machine learning algorithms, and artificial intelligence models, to estimate the future price movements of stocks.

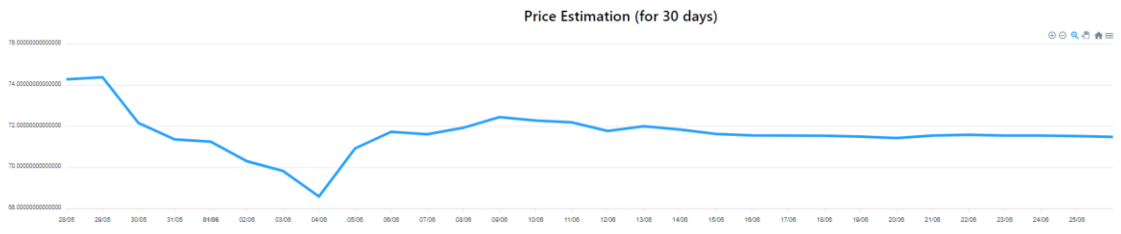


Fig-21:Prediction graph

4. Implementation and Testing

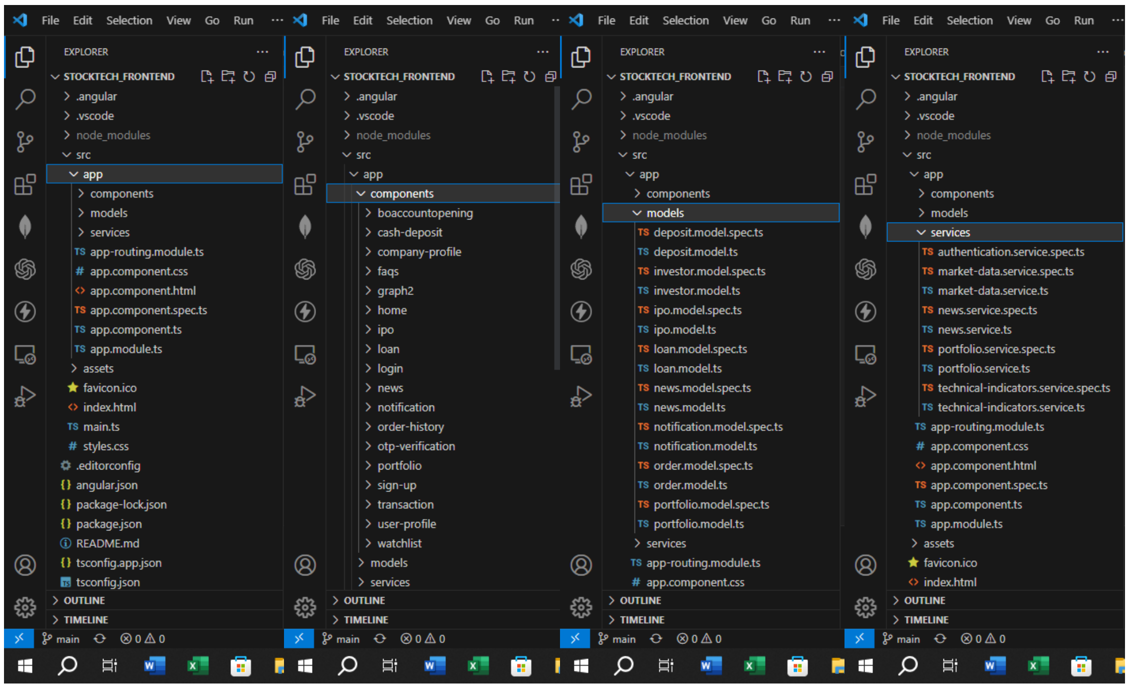
The process of implementing and testing involves gathering requirements, designing the system architecture, developing the front-end and back-end components, integrating databases, conducting thorough testing and quality assurance, deploying the website, conducting user acceptance testing, and maintaining the website for optimal performance and security. Throughout the process, attention is given to functionality, performance, user experience, and data integration to create a robust and user-friendly platform for stock market information and trading.

4.1. Implementation

The StockTech software is implemented using a combination of Angular for the front-end, Django for the back-end, MySQL for the database, Solidity for smart contracts, and LSTM for price prediction. The following details provide an overview of the implementation process:

4.1.1 Front-end Development (Angular)

StockTech's user interface is developed using the Angular framework, which provides a strong foundation for creating dynamic web applications. We use Angular components, services, and modules to build a responsive and interactive interface that users can easily navigate. To design and implement different features like charts and forms, we utilize HTML, CSS, and TypeScript. These technologies work together to create a user-friendly interface that allows users to interact with StockTech effectively.



```
STOCKTECH_FRONTEND
> .angular
> .vscode
> node_modules
v src
  v app
    > components
    > models
    > services
    app-routing.module.ts
    app.component.css
    app.component.html
    app.component.spec.ts
    app.component.ts
    app.module.ts
  > assets
  favicon.ico
  index.html
  main.ts
  styles.css
.editorconfig
angular.json
package-lock.json
package.json
README.md
tsconfig.app.json
tsconfig.json
```

```
STOCKTECH_FRONTEND
> .angular
> .vscode
> node_modules
v src
  v app
    v components
      > boaccountopening
      > cash-deposit
      > company-profile
      > faqs
      > graph2
      > home
      > ipo
      > loan
      > login
      > news
      > notification
      > order-history
      > otp-verification
      > portfolio
      > sign-up
      > transaction
      > user-profile
      > watchlist
    > models
    > services
```

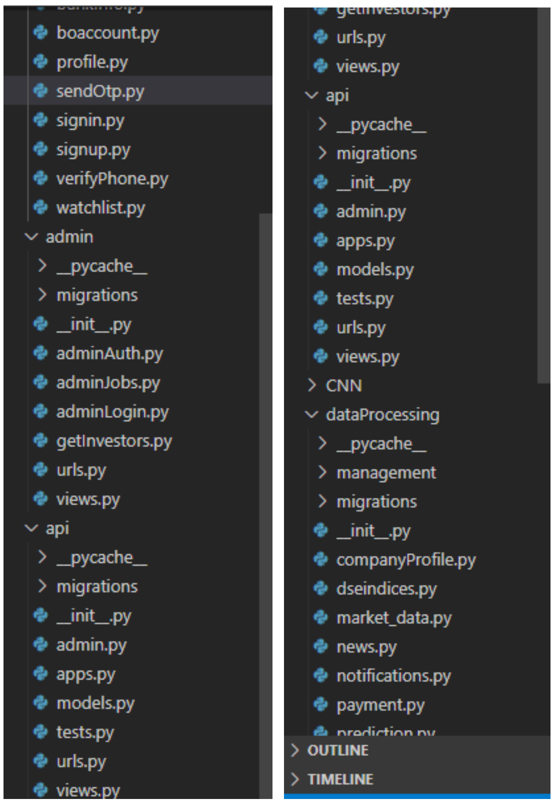
```
STOCKTECH_FRONTEND
> .angular
> .vscode
> node_modules
v src
  v app
    > components
    v models
      deposit.model.spec.ts
      deposit.model.ts
      investor.model.spec.ts
      investor.model.ts
      ipo.model.spec.ts
      ipo.model.ts
      loan.model.spec.ts
      loan.model.ts
      news.model.spec.ts
      news.model.ts
      notification.model.spec.ts
      notification.model.ts
      order.model.spec.ts
      order.model.ts
      portfolio.model.spec.ts
      portfolio.model.ts
    > services
    app-routing.module.ts
    app.component.css
```

```
STOCKTECH_FRONTEND
> .angular
> .vscode
> node_modules
v src
  v app
    > components
    > models
    v services
      authentication.service.spec.ts
      market-data.service.spec.ts
      market-data.service.ts
      news.service.spec.ts
      news.service.ts
      portfolio.service.spec.ts
      portfolio.service.ts
      technical-indicators.service.spec.ts
      technical-indicators.service.ts
    app-routing.module.ts
    app.component.css
    app.component.html
    app.component.spec.ts
    app.component.ts
    app.module.ts
  > assets
  favicon.ico
  index.html
```

Fig-22: Frontend Files

4.1.2 Back-end Development (Django)

The back-end of StockTech is created using Django, a robust web framework based on Python. Django's models, views, and serializers are utilized to manage data models, implement business logic, and define API endpoints. The back-end communicates with the front-end using RESTful APIs, enabling smooth data exchange between the two layers of the application. This ensures efficient handling of data and seamless integration between the back-end and front-end components of StockTech.



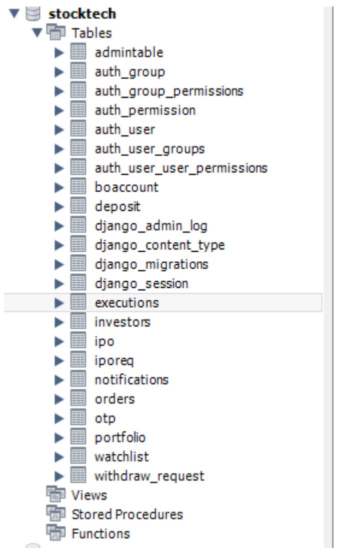
```
  boaccount.py
  profile.py
  sendOtp.py
  signin.py
  signup.py
  verifyPhone.py
  watchlist.py
v admin
  > __pycache__
  > migrations
  __init__.py
  adminAuth.py
  adminJobs.py
  adminLogin.py
  getInvestors.py
  urls.py
  views.py
v api
  > __pycache__
  > migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
```

```
  getInvestors.py
  urls.py
  views.py
v api
  > __pycache__
  > migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  urls.py
  views.py
> CNN
v dataProcessing
  > __pycache__
  > management
  > migrations
  __init__.py
  companyProfile.py
  dseindices.py
  market_data.py
  news.py
  notifications.py
  payment.py
> OUTLINE
> TIMELINE
```

Fig-23: Backend Files

4.1.3 Database Management (MySQL)

In StockTech, MySQL is employed as the database management system for storing and retrieving various types of data, including user accounts, portfolios, transactions, and other relevant information. Rather than utilizing Django's Object-Relational Mapping (ORM), direct SQL queries and statements are used to interact with the MySQL database. This approach provides greater flexibility and control over the database operations, enabling efficient data storage and retrieval within StockTech.

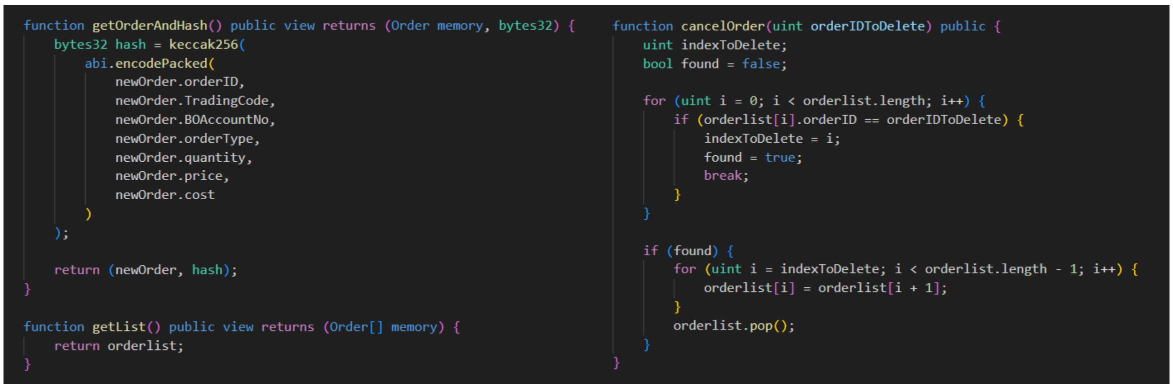


```
▼ stocktech
  ▼ Tables
    ▶ admintable
    ▶ auth_group
    ▶ auth_group_permissions
    ▶ auth_permission
    ▶ auth_user
    ▶ auth_user_groups
    ▶ auth_user_user_permissions
    ▶ boaccount
    ▶ deposit
    ▶ django_admin_log
    ▶ django_content_type
    ▶ django_migrations
    ▶ django_session
    ▶ executions
    ▶ investors
    ▶ ipo
    ▶ iporeq
    ▶ notifications
    ▶ orders
    ▶ otp
    ▶ portfolio
    ▶ watchlist
    ▶ withdraw_request
    Views
    Stored Procedures
    Functions
```

Fig-24: Database tables

4.1.4 Smart Contract Development (Solidity)

The smart contracts define the rules and conditions for executing trades, handling IPO applications, and managing the distribution of shares. In the process of implementing a smart contract, we utilized Solidity, a programming language specifically designed for writing smart contracts on the Ethereum blockchain.



```
function getOrderAndHash() public view returns (Order memory, bytes32) {
    bytes32 hash = keccak256(
        abi.encodePacked(
            newOrder.orderID,
            newOrder.TradingCode,
            newOrder.BOAccountNo,
            newOrder.orderType,
            newOrder.quantity,
            newOrder.price,
            newOrder.cost
        )
    );

    return (newOrder, hash);
}

function getList() public view returns (Order[] memory) {
    return orderlist;
}
```

```
function cancelOrder(uint orderIDToDelete) public {
    uint indexToDelete;
    bool found = false;

    for (uint i = 0; i < orderlist.length; i++) {
        if (orderlist[i].orderID == orderIDToDelete) {
            indexToDelete = i;
            found = true;
            break;
        }
    }

    if (found) {
        for (uint i = indexToDelete; i < orderlist.length - 1; i++) {
            orderlist[i] = orderlist[i + 1];
        }
        orderlist.pop();
    }
}
```

Fig-25: Code snippet- Order handling in the contract

To interact with the blockchain, we integrated Alchemy, a platform that provides APIs and infrastructure services. Metamask, a popular Ethereum wallet browser extension, was used to connect and interact with the blockchain network.

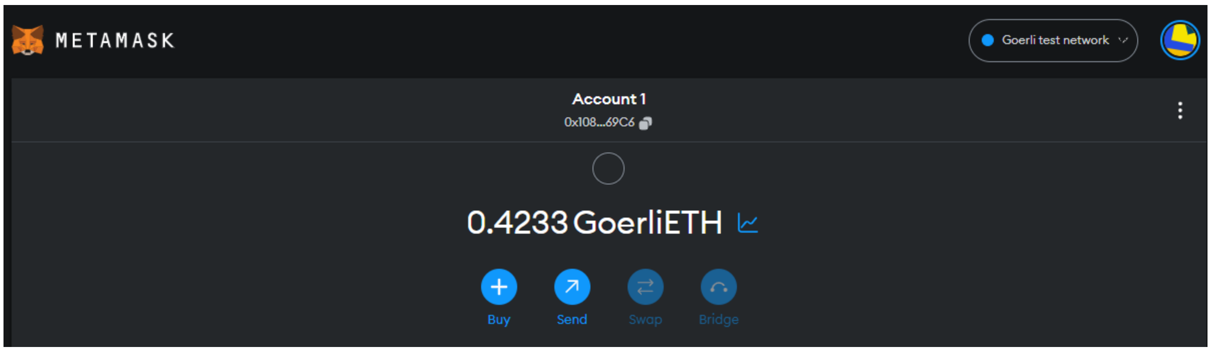
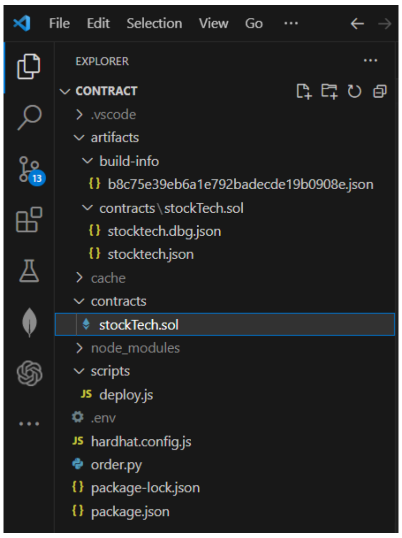


Fig-26: Metamask Account

For testing and development purposes, we employed the Goerli testnet, a test network that mimics the Ethereum mainnet environment but uses test Ether instead of real Ether. To facilitate the development and testing workflow, we utilized Hardhat, a development environment that offers built-in tools and scripts for smart contract deployment, testing, and debugging. This comprehensive process allowed us to develop, deploy, and test secure and efficient smart contracts on the Ethereum blockchain.

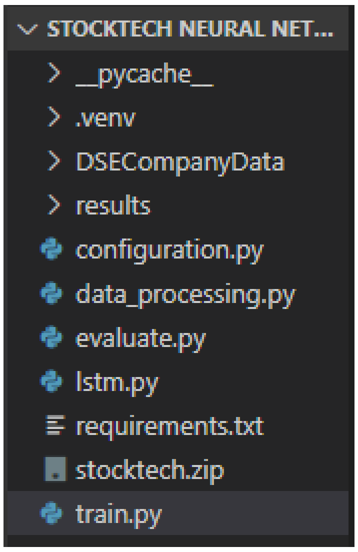


```
CONTRACT
  > .vscode
  v artifacts
    v build-info
      b8c75e39eb6a1e792badecde19b0908e.json
    v contracts\stockTech.sol
      stocktech.dbg.json
      stocktech.json
  > cache
  v contracts
    stockTech.sol
  > node_modules
  v scripts
    deploy.js
  .env
  hardhat.config.js
  order.py
  package-lock.json
  package.json
```

Fig-27: Contract files

4.1.5 Price Prediction (LSTM)

In StockTech, the LSTM (Long Short-Term Memory) algorithm plays a crucial role in price prediction. Historical stock market data is gathered and preprocessed to train LSTM models. Historical stock market data is collected, including information like historical prices, volume, and other relevant indicators. During the training process, the LSTM models learn to recognize and capture patterns within the data, enabling them to make predictions about future price movements. The models are designed to take into account both short-term and long-term dependencies, allowing them to capture complex relationships within the stock market data. Once the LSTM models are trained, they can be applied to new, unseen data to generate predictions. These predictions provide valuable insights to investors, assisting them in making informed decisions regarding their investments.



```
v STOCKTECH NEURAL NET...
  > __pycache__
  > .venv
  > DSECompanyData
  > results
  configuration.py
  data_processing.py
  evaluate.py
  lstm.py
  requirements.txt
  stocktech.zip
  train.py
```

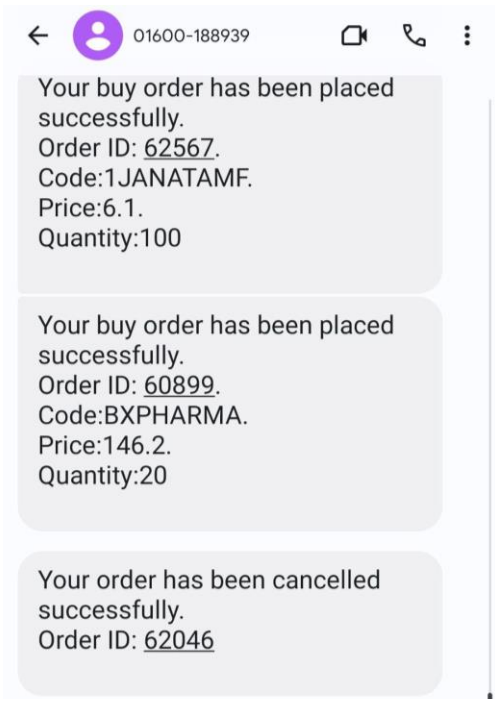
Fig-28: Tensor flow files

4.1.6 Admin front-end (Angular)

The Angular admin panel significantly enhances administrative control which is developed using the Angular framework. It is a powerful tool for managing withdrawal requests, notice announcements, IPOs, and investor-related data. The admin panel allows administrators to efficiently handle withdrawal requests by reviewing and processing them through a user-friendly interface.

4.1.7 Notification (SMS service)

The investor is notified for order confirmation, cancellation, execution, two factor authentication, account creation and any kind of transaction through automated SMS Service.


01600-188939

Your buy order has been placed successfully.
Order ID: 62567.
Code:1JANATAMF.
Price:6.1.
Quantity:100

Your buy order has been placed successfully.
Order ID: 60899.
Code:BXPHARMA.
Price:146.2.
Quantity:20

Your order has been cancelled successfully.
Order ID: 62046

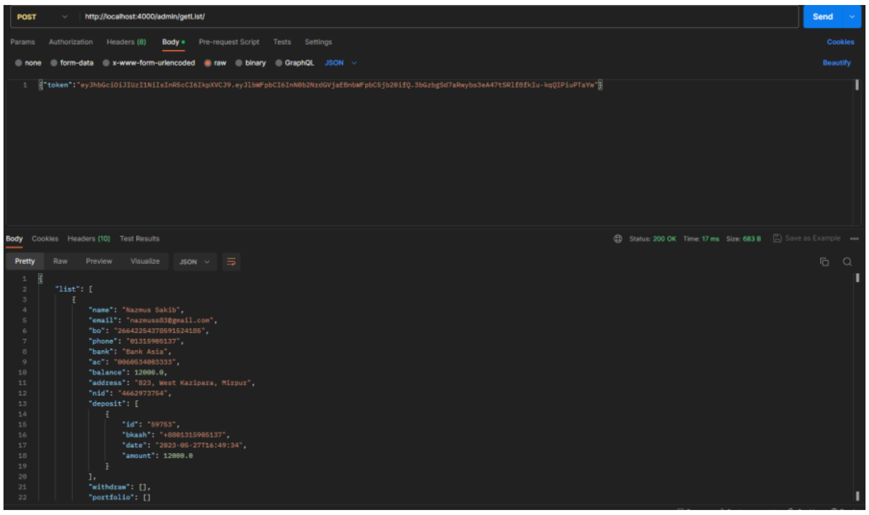
Fig-29: Notification through SMS

4.2. Testing

The StockTech software undergoes a comprehensive testing process to ensure its functionality, reliability, and security. The testing process includes various methods and tools to thoroughly evaluate different aspects of the application. Here is an overview of the testing process:

Unit Testing: Unit testing is performed to validate the correctness of individual components, functions, and modules within the software. These tests focus on isolating and testing small units of code to ensure they perform as expected.

Integration Testing: Integration testing is carried out to verify the seamless interaction and compatibility between different components of the StockTech software. This includes testing the integration between the front-end and back-end systems, ensuring proper data exchange and functionality. Tools like Postman and Thunder Client are utilized to send API requests and validate the responses from the back-end API endpoints.



http://localhost:4000/admin/getList/

```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFpbCI6InN0b2NrdGVjaEBnbWFpbC5jb20ifQ.3bGzbgSd7aRwybs3eA47tSRlf8fkIu-kqQIPiuPTaYw"}
```

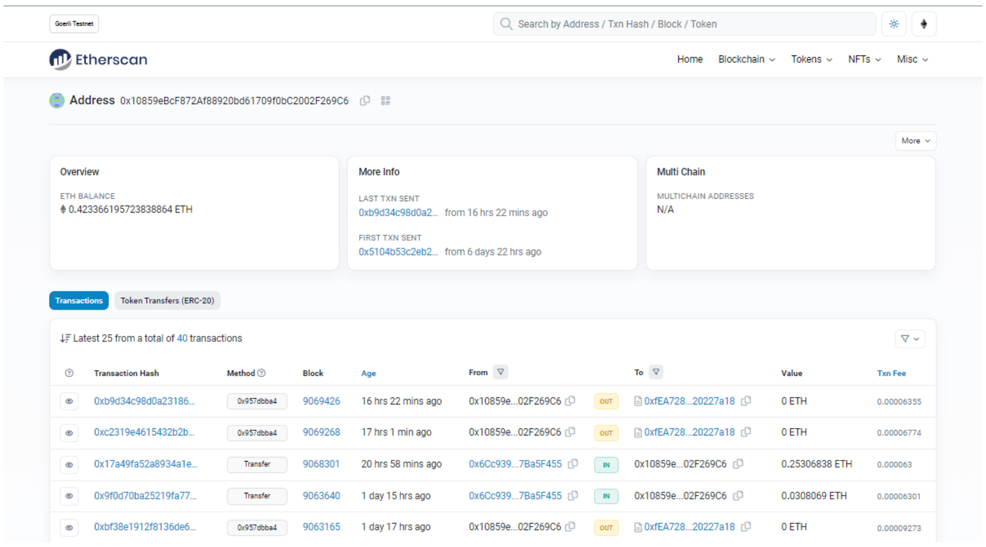
Status: 200 OK Time: 17 ms Size: 683 B

```
{
    "list": [
        {
            "name": "Nazmus Sakib",
            "email": "nazmuss83@gmail.com",
            "bo": "26642254378591524185",
            "phone": "01315905137",
            "bank": "Bank Asia",
            "ac": "0060534003333",
            "balance": 12000.0,
            "address": "823, West Kazipara, Mirpur",
            "nid": "4662973754",
            "deposit": [
                {
                    "id": "59753",
                    "bkash": "+8801315905137",
                    "date": "2023-05-27T16:49:34",
                    "amount": 12000.0
                }
            ],
            "withdraw": [],
            "portfolio": []
```

Fig-30: Testing using Postman

User Acceptance Testing: User acceptance testing involves engaging real users or test participants to evaluate the usability, user experience, and overall satisfaction with the StockTech application. Test participants perform specific tasks and provide feedback on the application's performance, ease of use, and any potential issues or improvements. This feedback is valuable for enhancing the software's design and functionality.

Transaction Testing: The StockTech software interacts with the Ethereum blockchain for transaction processing and smart contract execution. To test these functionalities, tools like Alchemy and the Goerli testnet are used. Alchemy provides a reliable infrastructure for connecting with the Ethereum network, while the Goerli testnet offers a sandbox environment for testing transactions without using real Ether. This ensures that the transactional features of StockTech are thoroughly tested and validated.



Address 0x10859eBcF872Af88920bd61709f0bC2002F269C6

Overview
ETH BALANCE
0.423366195723838864 ETH

More Info
LAST TXN SENT
0xb9d34c98d0a2... from 16 hrs 22 mins ago
FIRST TXN SENT
0x5104b53c2eb2... from 6 days 22 hrs ago

Multi Chain
MULTICHAIN ADDRESSES
N/A

Latest 25 from a total of 40 transactions

Transaction Hash	Method	Block	Age	From		To	Value	Txn Fee
0xb9d34c98d0a23186...	0x957dbba4	9069426	16 hrs 22 mins ago	0x10859e...02F269C6	OUT	0xfEA728...20227a18	0 ETH	0.00006355
0xc2319e4615432b2b...	0x957dbba4	9069268	17 hrs 1 min ago	0x10859e...02F269C6	OUT	0xfEA728...20227a18	0 ETH	0.00006774
0x17a49fa52a8934a1e...	Transfer	9068301	20 hrs 58 mins ago	0x6Cc939...7Ba5F455	IN	0x10859e...02F269C6	0.25306838 ETH	0.000063
0x9f0d70ba25219fa77...	Transfer	9063640	1 day 15 hrs ago	0x6Cc939...7Ba5F455	IN	0x10859e...02F269C6	0.0308069 ETH	0.00006301
0xbf38e1912f8136de6...	0x957dbba4	9063165	1 day 17 hrs ago	0x10859e...02F269C6	OUT	0xfEA728...20227a18	0 ETH	0.00009273

Fig-31: Goerli Etherscan

The testing process encompasses a combination of automated and manual testing techniques, allowing for comprehensive coverage of the StockTech software. Test plans, test cases, and bug tracking tools are utilized to manage and document the testing process, ensuring that all identified issues are properly addressed and resolved.

5. User Manual

The StockTech application is a user-friendly platform designed to assist investors in managing their stock market investments effectively. This user manual provides step-by-step instructions on how to utilize the various features and functionalities of the StockTech application.

We have two types of users-

- client
- admin

System Requirements:

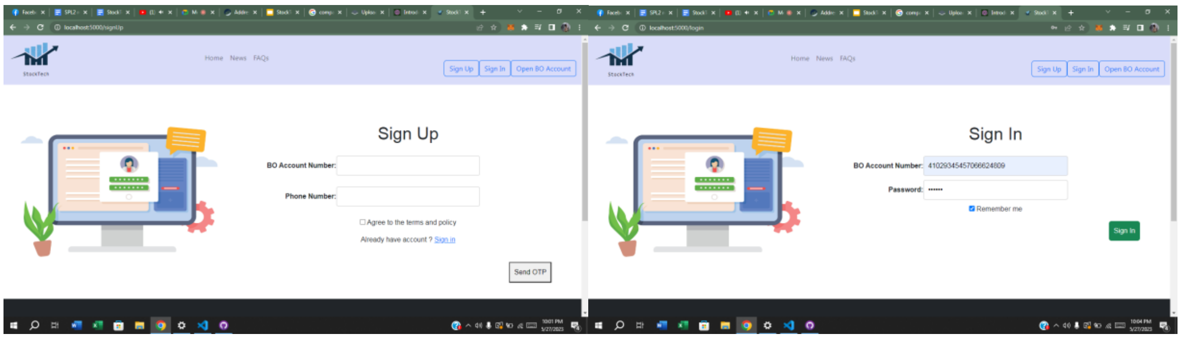
- Supported web browser (Chrome, Firefox, Safari, etc.)
- Internet connection

5.1 Client

5.1.1 Account Creation and Login

On the StockTech homepage, locate the "Sign Up" and "Open BO account" button.

To create a new BO account, fill in the required information and click the "Submit" button. To sign up, use the BO account number, phone number, password and create your StockTech account through OTP verification. Find the "Sign In" button. Enter your registered BO account number and password. Click on the "Login" button to access your StockTech account.

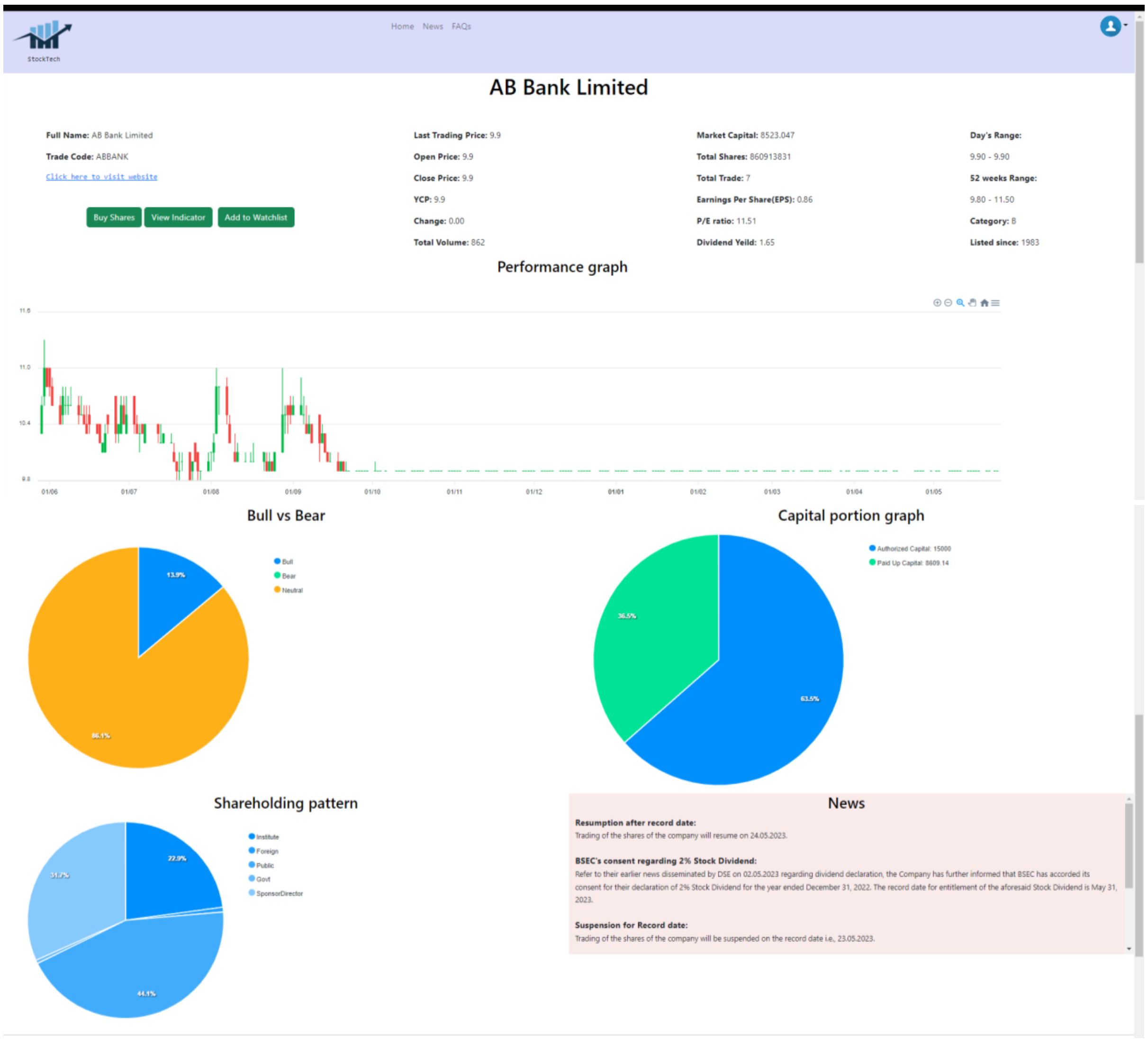


BO Account Number: 41029345457066624809
Password: ••••••
☑ Remember me

Fig-32: Account Creation and Login

5.1.2 Navigating the Dashboard

Upon successful login, you will be directed to the StockTech dashboard. The dashboard provides an overview of your stock market, different stock indices and company data. Utilize the navigation menu or tabs to access different sections of the application, such as portfolio, watchlist, transactions, news and order history. Explore the "News" section to access the latest market news, trends, and insights. Click on any company name to view the company profile. Click the "view graph" button to get analysis and recommendations.



AB Bank Limited

Full Name: AB Bank Limited
Trade Code: ABBANK
Click here to visit website

Last Trading Price: 9.9
Open Price: 9.9
Close Price: 9.9
YCP: 9.9
Change: 0.00
Total Volume: 862

Market Capital: 8523.047
Total Shares: 860913831
Total Trade: 7
Earnings Per Share(EPS): 0.86
P/E ratio: 11.51
Dividend Yeild: 1.65

Day's Range:
9.90 - 9.90
52 weeks Range:
9.80 - 11.50
Category: B
Listed since: 1983

News

Resumption after record date:
Trading of the shares of the company will resume on 24.05.2023.

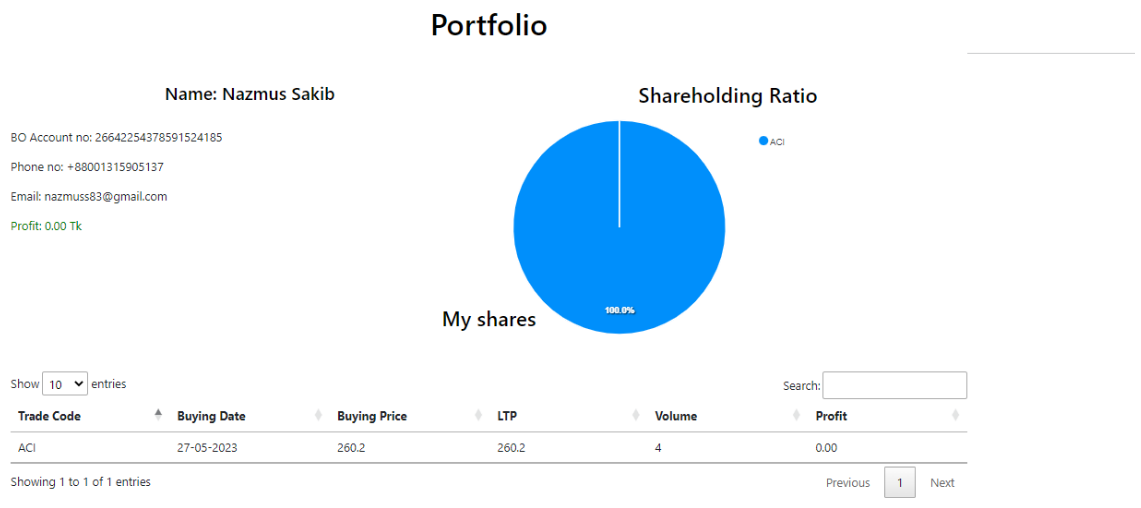
BSEC's consent regarding 2% Stock Dividend:
Refer to their earlier news disseminated by DSE on 02.05.2023 regarding dividend declaration, the Company has further informed that BSEC has accorded its consent for their declaration of 2% Stock Dividend for the year ended December 31, 2022. The record date for entitlement of the aforesaid Stock Dividend is May 31, 2023.

Suspension for Record date:
Trading of the shares of the company will be suspended on the record date i.e., 23.05.2023.

Fig-33: Company Profile Page

5.1.3 Managing Portfolio and watchlist

Click on the "Portfolio" tab to view and manage your stock holdings. The portfolio section displays your current holdings, including company names, quantities, and current market values. you can sell your holdings here. Click on the "Watchlist" tab to view the company data of your favorite companies.



Portfolio

Name: Nazmus Sakib

BO Account no: 26642254378591524185
Phone no: +88001315905137
Email: nazmuss83@gmail.com
Profit: 0.00 Tk

My shares

Show 10 entries Search:

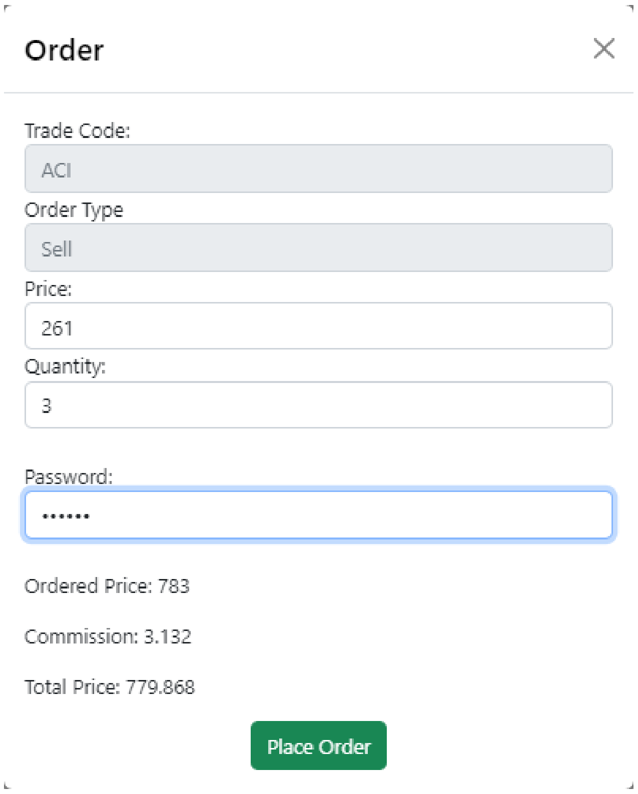
Trade Code	Buying Date	Buying Price	LTP	Volume	Profit
ACI	27-05-2023	260.2	260.2	4	0.00

Showing 1 to 1 of 1 entries Previous 1 Next

Fig-34: Portfolio

5.1.4 Placing an Order

To place an order for buying or selling stocks,click the "buy" button from the company. Enter the necessary details, such as the company name, quantity, and desired price. Review the order details and click on the "Place Order" button to submit your order. By clicking on the "apply" button in the IPO page, one can apply for IPOs.



Order

Trade Code: ACI

Order Type: Sell

Price: 261

Quantity: 3

Password: ••••••

Ordered Price: 783

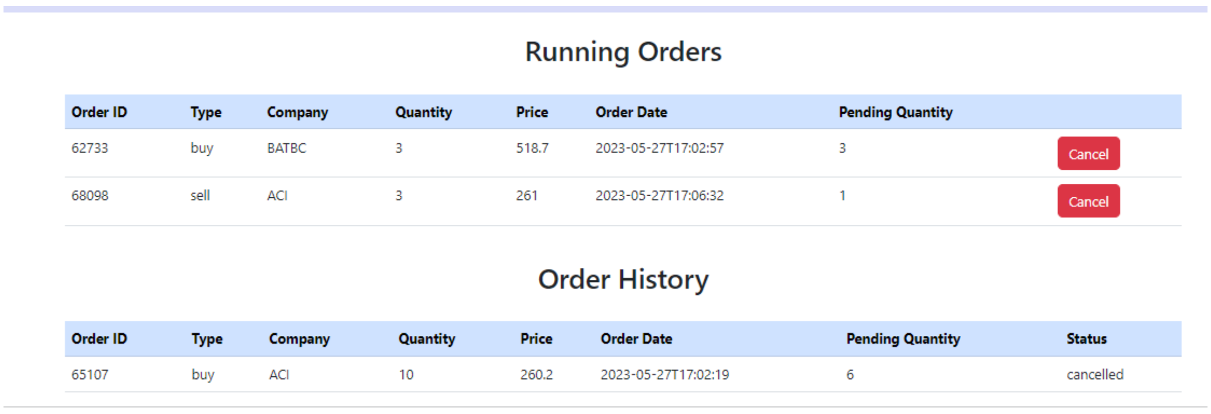
Commission: 3.132

Total Price: 779.868

Fig-35: Order Modals (buy, sell, ipo)

5.1.5 Check Order History

To check order history,click the "order history" button from the navigation bar. Click the "transaction history" button from the navigation bar to check previous transactions.



Running Orders

Order ID	Type	Company	Quantity	Price	Order Date	Pending Quantity	
62733	buy	BATBC	3	518.7	2023-05-27T17:02:57	3	Cancel
68098	sell	ACI	3	261	2023-05-27T17:06:32	1	Cancel

Order History

Order ID	Type	Company	Quantity	Price	Order Date	Pending Quantity	Status
65107	buy	ACI	10	260.2	2023-05-27T17:02:19	6	cancelled

Fig-36: Order History

For additional guidance and support, one can refer to the application's footer section to contact the StockTech support team.

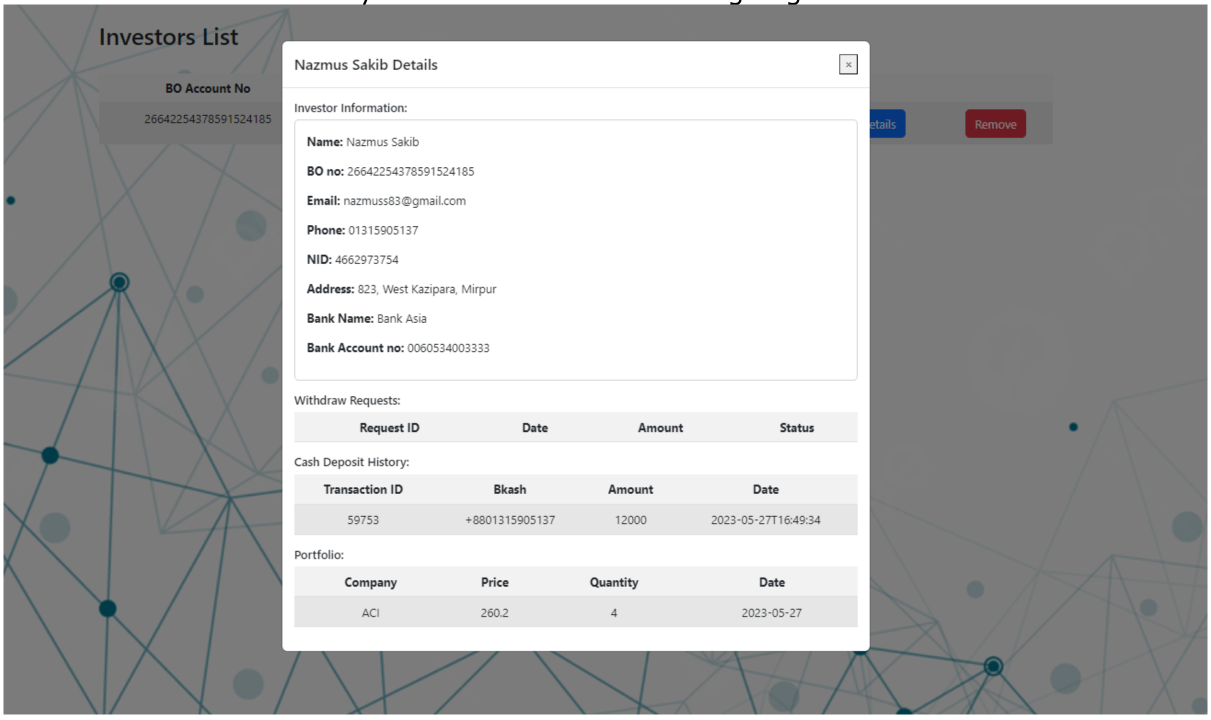
5.2 Admin

5.2.1 Authentication

Admin can access the system by providing credentials to the "sign in" page.

5.2.1 View Investor List

Admin can access the "Investor List" section to view all registered investors, investor details and can remove any investor if needed after signing in.



Nazmus Sakib Details

Investor Information:

Name: Nazmus Sakib
BO no: 26642254378591524185
Email: nazmuss83@gmail.com
Phone: 01315905137
NID: 4662973754
Address: 823, West Kazipara, Mirpur
Bank Name: Bank Asia
Bank Account no: 0060534003333

Withdraw Requests:

Request ID	Date	Amount	Status

Cash Deposit History:

Transaction ID	Bkash	Amount	Date
59753	+8801315905137	12000	2023-05-27T16:49:34

Portfolio:

Company	Price	Quantity	Date
ACI	260.2	4	2023-05-27

Fig-37: Investor list page

5.2.2 Add Announcements

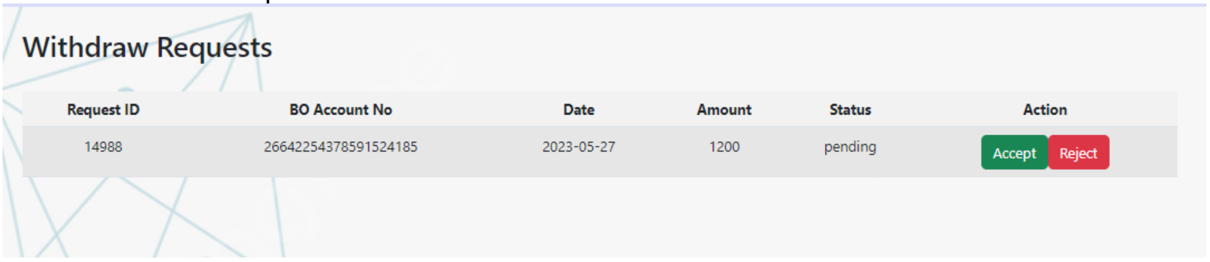
Compose and publish important announcements in the "Add Announcement" section to keep investors informed.

5.2.3 IPO List

Admin can track ongoing and upcoming IPOs in the "IPO List" section, including company names and pricing information. Admin can also announce a new IPO to the investors. He can also make IPO announcements by providing the relevant details by clicking the "Announce IPO" button.

5.2.4 Withdrawal Requests

Admin can monitor and manage investor withdrawal requests in the "Withdrawal Requests" section. Here he can view withdrawal requests and either approve or reject them based on the provided information.


Withdraw Requests

Request ID	BO Account No	Date	Amount	Status	Action
14988	26642254378591524185	2023-05-27	1200	pending	Accept Reject

Fig-38: Withdrawal page

6. Challenges Faced

The development and implementation of StockTech came with its fair share of challenges. From data quality and availability to model complexity and performance, several hurdles had to be overcome. Additionally, the dynamic and non-linear nature of stock markets, the need for computational resources, and ensuring interpretability and compliance added to the complexities. During the development and implementation of StockTech, several challenges were encountered such as-

6.1 Technical Complexity

Developing a comprehensive stock market application involved working with various technologies and integrating different components such as frontend frameworks, backend systems, databases, and smart contracts. The technical complexity posed a challenge in terms of system architecture, integration, and ensuring smooth functionality.

Overcoming: We conducted thorough research and analysis to understand the technical requirements and selected appropriate technologies for each component. We utilized documentation and online resources and code reviews that helped us identify and resolve issues promptly.

6.2 Data Integration and Accuracy

Accurate and up-to-date stock market data is crucial for the functioning of the application. We faced challenges in integrating data feeds from multiple sources, ensuring data accuracy, and handling real-time updates to provide users with reliable information.

Overcoming: We leveraged reliable data providers and APIs to integrate real-time market data into our application. Rigorous testing and monitoring were performed to address any discrepancies or data inconsistencies.

6.3 Security and Transaction Safety

The stock market involves financial transactions, and ensuring the security and safety of user transactions was a significant challenge. Building a secure system that protects user data, prevents unauthorized access, and safeguards against fraudulent activities requires careful planning and implementation.

Overcoming: We included implementing encryption techniques, incorporating secure authentication mechanisms, and adhering to best practices in data handling and storage.

By actively addressing these challenges, we were able to overcome them and deliver a robust and user-friendly StockTech application. Our continuous improvement efforts played a vital role in successfully navigating and resolving these challenges.

7. Conclusion and Future Work

Throughout the development of the StockTech application, we gained valuable insights and experiences. We learned the intricacies of integrating diverse technologies such as Angular for the frontend, Django for the backend, MySQL for the database, and Solidity for smart contracts. The StockTech application has provided investors with a user-friendly and efficient platform for managing their stock market investments. It has successfully addressed challenges such as technical complexity, data integration, security, and user adoption. By leveraging blockchain technology, we have enhanced transaction security and transparency, fostering trust and confidence among users.

Expansion to Additional Financial Markets: The StockTech application can be extended to support other financial markets, such as commodities, bonds, or foreign exchange. This expansion will provide users with a broader range of investment options and diversification opportunities.

Implementation of Social Trading Features: Introducing social trading features, where users can follow and replicate the investment strategies of successful traders, can enhance the collaborative aspect of the platform. This will enable users to learn from each other and potentially improve their investment performance.

Mobile Application Development: Developing a mobile application for StockTech will provide users with the convenience of accessing their investment portfolios and executing trades on their smartphones. This expansion will cater to the growing mobile-centric user base and increase accessibility.

Enhanced User Personalization: Introducing personalized dashboards and tailored investment recommendations based on user preferences and risk profiles can enhance the user experience. This personalization will cater to individual investment goals and strategies.

Sentiment Analysis for Predictive Insights: Incorporating sentiment analysis techniques can provide valuable predictive insights to investors. By analyzing social media feeds, news articles, and other textual data, the application can gauge market sentiment and sentiment towards specific stocks. This information can be integrated into the analysis tools to offer users a holistic view of market sentiment and potential impact on stock prices.

By pursuing these avenues, the StockTech application can continue to evolve and provide users with an advanced and comprehensive platform for managing their investments. The lessons learned from the project will guide future development, ensuring the application remains at the forefront of technological advancements in the financial industry.

Reference

- Guide, Step, and Shipra Saxena. "LSTM | Introduction to LSTM | Long Short Term Memory Algorithms." *Analytics Vidhya*, 16 March 2021, https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/. Accessed 27 May 2023
- Graves, Alex. "Understanding LSTM Networks -- colah's blog." *Colah's Blog*, 27 August 2015, https://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed 27 May 2023.
- Pramod, & Pm, Mallikarjuna. (2021). Stock Price Prediction Using LSTM. Test Engineering and Management. 83. 5246-5251.
- "Developing smart contracts." *OpenZeppelin Docs*, https://docs.openzeppelin.com/learn/developing-smart-contracts. Accessed 27 May 2023.
- "Intro to Smart Contracts." *Web3 University*, https://www.web3.university/tracks/create-a-smart-contract. Accessed 27 May 2023.
- "solidity - Execute a contract function from web3.py." *Ethereum Stack Exchange*, 31 August 2022, https://ethereum.stackexchange.com/questions/134720/execute-a-contract-function-from-web3-py. Accessed 27 May 2023.
- *gm — web3.py 6.4.0 documentation*, https://web3py.readthedocs.io/en/stable/. Accessed 27 May 2023.

Appendix

Goerli Faucet

- https://faucet.quicknode.com/ethereum/goerli
- https://goerlifaucet.com/

SMS Service: https://www.greenweb.com.bd/
Data Resource: https://www.amarstock.com/

Github: StockTech
SRS: StockTech SRS