Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B. Tech. CE Semester - V

Subject: (CE-520) Advanced Technologies

Project Title: Stalk Your Stock - Trading Simulator WebApp

Prepared By:

Name: Hitarth Patel **Roll No:** CE098 **ID:** 21CEUOG047 **Batch:** B2	**Name:** Neer Patel **Roll No:** CE103 **ID:** 21CEUOS152 **Batch:** B2

Guided by: Prof. Ankit P. Vaishnav

Certificate

This is to certify that the project entitled “Stalk Your Stock”, in the subject of Advanced Technology is a Bonafede report of the work carried out by

Patel Hitarth Nileshbhai (CE098) (21CEUOG047)

Patel Neer Hasmukhbhai (CE103) (21CEUOS152)

Of B.Tech semester V in the Branch of Computer Engineering

during the Academic Year 2023-2024

Prof. Ankit P. Vaishnav
Dept of Computer Engg.
Faculty of Technology

Dr. C. K. Bhensdadia
Head of Dept of Computer Engg.
Faculty of Technology

Content

Abstract

"Stalk Your Stock" is an innovative web-based stock market simulator, designed to provide users with a risk-free environment to practice trading skills and gain insights into the financial markets. This project report offers a detailed overview of the platform, covering user registration, real-time stock data, portfolio management, interactive charts, and more.

The project aims to empower users with virtual credits, allowing them to simulate buying and selling stocks and enhancing their understanding of the stock market and trading strategies. Leveraging the MERN stack and integrating with the Yahoo Finance API, "Stalk Your Stock" is a standalone web-based platform.

This report is tailored for developers, project managers, marketing teams, potential users, testers, and documentation writers, offering insights for each stakeholder group.

The platform operates under the assumption that users possess basic stock trading knowledge and rely on the Yahoo Finance API for real-time data. It prioritises security, scalability, and performance.

In summary, "Stalk Your Stock" is an educational and user-friendly tool that fosters financial literacy and provides hands-on experience in stock trading, all within a secure, responsive environment.

Introduction

- ## Purpose

  The "Stalk Your Stock" project is dedicated to developing a sophisticated web-based platform for stock trading enthusiasts. By harnessing the power of the MERN stack and seamlessly integrating with the Yahoo Finance API, it provides users with a secure, simulated environment to enhance their trading skills, formulate effective strategies, and bolster confidence before venturing into real-world trading. This initiative not only fosters financial literacy but also empowers users with the knowledge and experience necessary for navigating the complexities of the stock market.

- ## Overall Description

  "Stalk Your Stock" is a dynamic web-based platform tailored to empower both novice and experienced stock market enthusiasts. In the ever-evolving world of financial markets, managing stock portfolios and making well-informed investment decisions can be a daunting task. This platform has been meticulously designed to simplify stock portfolio management, market analysis, and decision-making processes, offering an accessible solution for all levels of experience.

  A cornerstone of this platform is its integration with the Yahoo Finance API, providing real-time stock market data to keep users informed about market trends. The user-friendly interface makes

stock market analysis intuitive, ensuring that even newcomers can navigate the complexities of stock trading. For experienced traders, "Stalk Your Stock" offers a sandbox for refining investment strategies and tactics without real financial risk.

In a world where financial markets are increasingly intricate, "Stalk Your Stock" stands as a guiding light for traders. It works seamlessly across devices and browsers, providing a responsive and seamless user experience with minimal downtime. Security, usability, and data integrity are paramount. The platform employs robust security measures, offers a user-friendly interface for individuals of varying experience, and maintains data accuracy.

"Stalk Your Stock" is not just a web app; it's your partner in the world of stock trading. It simplifies the complexities of the stock market, provides invaluable tools for analysis, and assists users in making informed investment decisions. Whether you're an experienced trader or a newcomer, "Stalk Your Stock" is your companion in stock portfolio management and market analysis.

- **Tools/Technologies Used**

The development of "Stalk Your Stock" involved a set of tools and technologies carefully selected to ensure efficiency and effectiveness throughout the project. These tools and technologies were instrumental in creating a robust and user-friendly web application:

Development Stack

- **Frontend:**
    - **Next.js:** Used as the core frontend framework for building the UI.
    - **Tailwind CSS:** Employed for structuring and styling web pages.
    - **Particles.js:** Module used for unique and eye-catching background UI.
- **Backend:**
    - **Node.js**: The runtime environment for the backend server.
    - **Express.js:** A web application framework for Node.js, for routing and making middleware.
- **Database:**
    - **MongoDB:** A NoSQL database for storing and managing data efficiently.
    - **Mongoose:** Module used for object-data modelling

API

- **Yahoo Finance:** Used for fetching real-time stock prices and data.
- **Axios:** Used for fetching data from APIs

Data Visualization

- **Chart.js:** Used for creating graphical analytics to visualize stock prices over history.

Version Control

- **Git/GitHub:** Used for version control and collaboration among project contributors.

Code Editor

- **Visual Studio Code (VS Code)**: The primary code editor for development and debugging.

These tools and technologies, thoughtfully selected, played a vital role in shaping the "Stalk Your Stock" project into a functional and user-friendly web application.

Software Requirement Specifications(SRS)

1. Introduction:

1.1. Purpose:

The purpose of the Stock Market Simulator project is to create a web-based platform that offers users a risk-free environment to practice and improve their trading skills. By utilizing the MERN stack and integrating it with the Yahoo Finance API, the platform allows users to simulate buying and selling stocks using virtual credits and provides real-time data from the stock market. Through this virtual trading experience, users can enhance their understanding of the stock market, develop effective trading strategies, and gain confidence before venturing into real-world trading.

1.2. Intended Audience and Reading Suggestions:

The intended audience for this SRS document of the Stock Market Simulator project includes developers, project managers, marketing staff, potential users, testers, and documentation writers. Developers can refer to the technical specifications and architecture details in the document. Project managers can gain insights into project scope and resource requirements. Marketing staff can understand the platform's features for promotional purposes. Potential users can explore the functionality and benefits. Testers can refer to the

requirements for validation. Documentation writers can extract information from user manuals.

1.3. Product Scope:

The Stock Market Simulator is a web-based software designed to provide users with a risk-free environment for practising trading skills and analyzing the stock market. The platform aims to empower users with virtual credits to simulate buying and selling stocks, enabling them to enhance their understanding of the stock market, develop effective trading strategies, and gain confidence before venturing into real-world trading. The software aligns with the objective of creating an educational and interactive tool to foster financial literacy and empower users with valuable hands-on experience in stock market trading.

1.4. References:

1.4.1. NextJS, ReactJS, NodeJS, Tailwind CSS, npm documentation

1.4.2. Yahoo Finance

1.4.3. Stack Overflow

2. General Description:

2.1. Product Perspective:

The Stock Market Simulator is a new, self-contained web-based platform, not part of an existing product family or a replacement for any system. Operating independently, it offers users a risk-free environment to practice trading skills and analyze the stock market. It interacts with the Yahoo Finance API for real-time data but functions autonomously without external dependencies.

2.2. Product Functions:

2.2.1. Functionality of the system

- Easy User Registration/Login
- Virtual Credits and Trading Simulation
- Real-time Stock Data
- Stock Search and Analysis
- Portfolio Management
- Transaction Management
- Interactive Charts
- User-friendly and Intuitive UI

2.2.2. User classes and Characteristics:

- **Researchers:** Researchers will be coming to the website for live stock prices and past charts to do technical analysis.
- **Trainees:** They will come to the website for the training of the stock market using our virtual credits.

2.2.3. Operating Environment:

The system will be compatible with all popular web browsers, including Chrome, Firefox, Safari, Edge, and Internet Explorer. The machine should have sufficient speed of internet connection.

2.2.4. Design and Implementation Constraints:

The app will be designed using modern web technologies and frameworks ensuring cross-platform compatibility.

2.2.5. Assumptions and Dependencies

- **Assumptions**:
    - The availability and reliability of the Yahoo Finance API to provide real-time stock market data.
    - Users have a basic understanding of stock trading concepts, as the simulator assumes some level of prior knowledge.
    - The web-based platform will be accessible to users on common browsers and devices with standard internet connectivity.
    - Users will engage responsibly and abide by ethical guidelines while using the virtual trading environment.

- **Dependencies**:

- The successful integration with the Yahoo Finance API for obtaining live stock market data.
- Dependence on the MERN stack (MongoDB, Express.js, React.js, Node.js) for the development and functioning of the web-based platform.
- The availability of third-party libraries or components used for data analysis, charting, and visualization within the platform.

3. Specific Requirements:

3.1. External Interface Requirements:

3.1.1. User Interfaces:

- **Registration and Login:** A user-friendly registration and login system to facilitate account creation and secure user authentication.
- **Profile Page:** A comprehensive user profile page allowing users to manage their personal information, preferences, and settings.
- **Portfolio Page:** An intuitive interface for users to create, edit, and monitor their stock portfolios, providing an overview of their investments.

- **Transaction Page:** A dedicated section to review and track transaction history, offering insights into stock buying and selling activities.
- **Stock Search and Buy/Sell Page:** An interactive platform enabling users to search for specific stocks, assess their performance, and execute buy/sell orders.
- **Stock Chart Page:** A dynamic charting interface that offers graphical representations of stock market data, aiding users in analyzing market trends and making informed decisions.

3.1.2. Hardware Interfaces

The system is compatible with a wide range of hardware, including desktop computers, laptops, and mobile devices. The system will be able to run on popular operating systems such as Windows, MacOS, and Linux.

3.1.3. Software Interfaces

The database connection interface is part of the Stock market analysis and virtual trading platform that manages the communication and interaction between the system and the underlying database.

3.1.4. Communication Interfaces

This uses standard network protocols, such as HTTPS. The system will comply with industry standards for data encryption and secure communication to ensure that customer information and orders are protected from unauthorized access or attack.

3.2. Functional Requirements:

R.1: User Registration and Authentication

R.1.1: User Registration

Description: The system should allow users to create an account by providing a unique username and password.

Input: The user provides a unique username and password.

Output: The user account is created and stored securely in the system.

R.1.2: User Login

Description: The system should provide a login mechanism to authenticate registered users based on their credentials.

Input: The user enters their username and password.

Output: The user is successfully authenticated and granted access to their account.

R.2: Virtual Trading

R.2.1: Virtual Credits Allocation

Description: The system should allocate virtual credits to each registered user upon account creation.

Input: User account is created.

Output: The user is assigned a predefined amount of virtual credits.

R.2.2: Stock Trading Simulation

Description: The system should allow users to simulate buying and selling stocks using the provided virtual credits.

Input: The user selects stocks to buy or sell and specifies the quantity.

Output: The user's virtual credits are adjusted based on the simulated transaction.

R.3: Show real-time stock prices and data

Description: The system should integrate with the Yahoo Finance API to fetch live data from the stock market.

Input: The system fetches data from the Yahoo Finance API.

Output: The system receives real-time stock market data to display to users.

R.4: Stock search and analysis

R.4.1: Stock Search

Description: The system should provide search functionality for users to find specific stocks based on their names or symbols.

Input: The user enters the stock name or symbol in the search bar.

Output: The system displays search results matching the entered criteria.

R.4.2: Stock Analysis

Description: The system should retrieve detailed stock information, including stock price, volume, market trends, and other relevant indicators.

Input: The user selects a specific stock for analysis.

Output: The system displays detailed stock information to the user for analysis.

R.5: Manage Portfolio

Description: The system should enable users to create and manage multiple investment portfolios associated with their accounts.

Input: The user creates or modifies a portfolio, and adds or removes stocks.

Output: The user's portfolios are updated with the changes made.

R.6: Manage Transaction History

R.6.1: Show transaction History

Description: The system should maintain a transaction history for each user's portfolio, recording buy/sell actions and corresponding virtual credits.

Input: User performs buy/sell actions on stocks.

Output: The user's transaction history is updated to reflect the actions taken.

R.7: Show interactive Charts

Description: The system should provide interactive charts to visualize stock price trends, volume, and other key indicators.

Input: The user selects the desired stock and chart options.

Output: The system displays interactive charts with stock market data to the user

3.3. Non-functional Requirements:

3.3.1. Performance: The platform should provide a responsive and smooth user experience, ensuring fast loading times and quick execution of virtual trades.

3.3.2. Security: The system should employ strong security measures, including encryption, secure authentication, and access controls, to protect user data and transactions.

3.3.3. Usability: The user interface should be intuitive and user-friendly, with clear navigation and well-organized

features to accommodate users of varying experience levels.

3.3.4. **Reliability:** The platform should be highly reliable, minimizing downtime and ensuring continuous availability to avoid disruptions during critical trading periods.

3.3.5. **Data Integrity:** The platform should maintain data accuracy and consistency, ensuring that virtual trades, portfolio information, and historical data remain reliable and error-free.

Database Design

- **Database collections**:
    - User Schema:
        - fname: The user's given name or forename.
        - lname: The user's family name or surname.
        - email: The user's unique email address for communication and account verification.
        - phone: The user's contact number for notifications and account recovery.
        - city: The city where the user resides.
        - state: The state or region within the user's country.
        - country: The user's home country.
        - pincode: The postal code associated with the user's address.
        - address: The physical location where the user lives, including street, apartment, or house details.
        - password: A secure string of characters used to protect the user's account.
        - credit: The amount of virtual currency or credits associated with the user's account for stock trading simulations.
        - transaction: An array of primary keys of transactions done by the user.

- Portfolio Schema:
  - stockName: Name of the company whose stocks are being purchased.
  - currentBuyings: Overall value of purchases made for the current company.
  - currentQuantity: Total number of shares acquired for the present company.
  - user: Foreign key references to the user's ID.
- Transaction Schema:
  - stockName: Name of the company whose stocks are being purchased.
  - isBuy: Boolean value indicating whether the shares were purchased (True) or sold (False).
  - price: The rate at which a transaction for a specific amount has been executed.
  - time: The time at which a transaction has been executed.
  - quantity: The quantity of the share for the transaction.
  - user: Foreign key references to the user's ID.

- **Relationships:**
    - User and Portfolio: User and Portfolio is one-to-many, meaning that a single user can have multiple portfolio entries, each corresponding to a different company.
    - User and Transaction: User and Transaction is one-to-many, signifying that a single user can engage in multiple transactions.

Implementation Details

- **Module: userController.js**

  The "userController.js" module is an essential component of the "Stalk Your Stock" application, primarily focusing on user-related functionalities, including registration, login, profile management, and logout. It ensures user data security and provides a seamless experience for app users.

  Function Descriptions:

  1. securePassword(password):

     - Description: This function secures user passwords using SHA-256 hashing to protect sensitive data. It enhances the application's security.

  2. login(req, res):

     - Description: The "login" function manages user login attempts, validating email and password. It grants access to authenticated users and handles various login scenarios.

  3. register(req, res):

     - Description: The "register" function enables user registration, Create a new user profile with essential details. It enhances

the application's user base and ensures data integrity and security.

4. logout(req, res):

- Description: The "logout" function handles user logouts, ensuring a secure and smooth exit from the application.

5. getProfile(req, res):

- Description: The "getProfile" function retrieves user profile data based on the provided email. It ensures users can access and update their profile information as needed.

6. updateProfile(req, res):

- Description: The "updateProfile" function allows users to modify and update their profile information. It ensures user data accuracy and personalization of profiles.

- **Module: portfolioController.js**

The "portfolioController.js" module is a critical component of the "Stalk Your Stock" application, responsible for managing user portfolios and their related actions. It facilitates the addition of stocks to a user's portfolio and retrieval of portfolio data.

Function Descriptions:

1. addToPortfolio(req, res):

    - Description: The "addToPortfolio" function allows users to add stocks to their portfolio, specifying stock details like name, quantity, and current price. It adds the stock to the user's portfolio and reflects the corresponding changes in their holdings and credit.

2. getPortfolio(req, res):

    - Description: The "getPortfolio" function retrieves a user's portfolio, including all the stocks they currently own. It provides a comprehensive view of the user's investments and their available credit balance.

- **Module: transactionController.js**

    The "transactionController.js" module is a core element of the "Stalk Your Stock" application, primarily handling user transactions, stock purchases, and sales. It orchestrates these actions, making sure that the user's portfolio and credit are updated accurately in response to their transactions.

    Function Descriptions:

1. addTransaction(req, res):

    - Description: The "addTransaction" function manages the addition of stock transactions. It considers factors like stock name, quantity, current price, and transaction type (buy/sell).

This function updates the user's portfolio and credit balance accordingly and stores the transaction data for record-keeping.

2. getTransaction(req, res):

- Description: The "getTransaction" function retrieves a user's transaction history. It presents the transaction data in reverse chronological order to offer users an overview of their recent buying and selling activities.

- **Module: stockInfoController.js**

The "stockInfoController.js" module is responsible for providing users with essential information regarding stock names and their corresponding details. It interacts with external APIs and local data to fetch and present this data.

Function Descriptions:

1. nameInfo(req, res):

- Description: The "nameInfo" function reads stock name information from a local file (EQUITY_L.csv). It processes this information and extracts stock names. The function then sends this parsed data as a response, making it accessible to users.

2. stockInfo(req, res):

- Description: The "stockInfo" function retrieves detailed information about a particular stock using its symbol. It queries

Yahoo Finance API for financial data related to the provided stock symbol and sends the details as a response. If the stock symbol is not found or if there's an internal server error, appropriate error messages are sent.

- **Module: chartController.js**

The "chartController.js" module serves a pivotal role in the "Stalk Your Stock" application by facilitating the retrieval of historical stock price data and formatting it for user consumption. This data is then presented graphically in the form of stock price charts.

Function Descriptions:

1. chart(req, res):

- Description: The "chart" function fetches historical stock price data from Yahoo Finance API based on the provided stock symbol. It specifies the desired metrics and time intervals for the chart. The function processes the retrieved data, converting timestamps to human-readable dates and formatting stock prices. It sends the formatted data as a response to enable users to visualize historical stock price trends graphically.

Testing

The "Stalk Your Stock" project is underpinned by a robust testing framework and methodology to ensure the reliability and functionality of this web-based stock market simulator. The testing process combines manual and automated testing techniques and is designed to validate various aspects of the platform.

- **Testing Framework:**
    - **Manual Testing:** Manual testing is an integral part of our framework and involves human testers interacting with the system to ensure that it meets user expectations, identify UI/UX issues, and validate core functionalities.
    - **Automated Testing:** Automated testing is conducted using well-established testing frameworks (Eg. Selenium tool) to execute test cases and verify the system's functionalities efficiently.

- **Testing Methods:**
    - **Unit Testing:** This method focuses on the testing of individual units or components of the system, ensuring they function correctly. For instance, we conduct unit tests on individual API endpoints to validate data retrieval, updates, and deletions.
    - **Integration Testing:** Integration testing examines how different components or modules of the system interact with each other. We rigorously test scenarios such as user

registration and role assignment to ensure seamless integration.

- **Functional Testing:** Functional testing is crucial to validate the overall functionality of the platform. This includes core features such as user registration, stock trading, portfolio management, and interactive chart functionalities.
- **UI/UX Testing:** Our team performs manual testing to assess the user interface and user experience. Testers evaluate the system's responsiveness, design, and ease of use to ensure an intuitive UI.
- **Security Testing:** Security testing is a key aspect of our framework, where we actively identify vulnerabilities and validate data protection measures to maintain the security of user data and transactions.

- **Sample Test Cases:**

  - **Unit Test (Backend - Node.js/Express):**
    Test Case 1: Verify that a new user can be successfully created with valid input data.
    Test Case 2: Ensure that the system returns the correct user data when querying by user ID.

  - **Integration Test (User and Role Management):**
    Test Case 3: Validate that a user's role is correctly assigned after registration.
    Test Case 4: Verify that an admin can update user roles and permissions successfully.

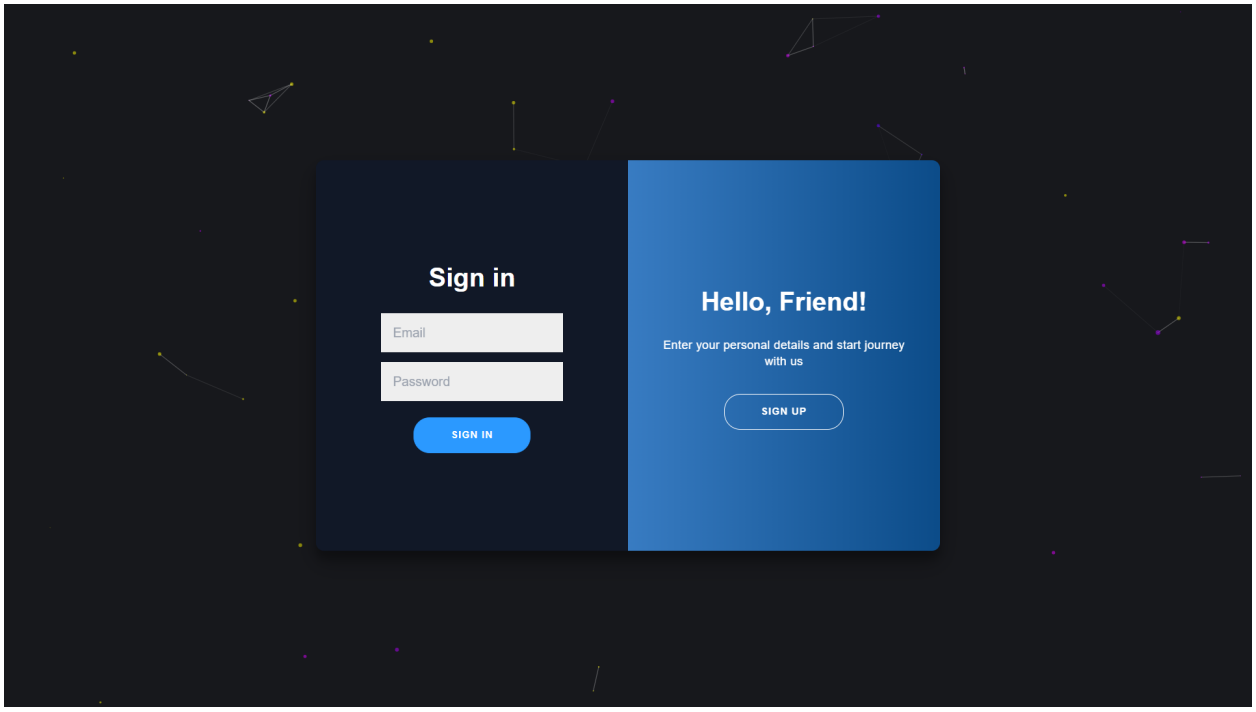
- **Functional Test (Stock Trading):**
  Test Case 5: Check if the system allows users to successfully buy and sell stocks using virtual credits.
  Test Case 6: Confirm that the user's virtual credits are adjusted accurately after stock transactions.

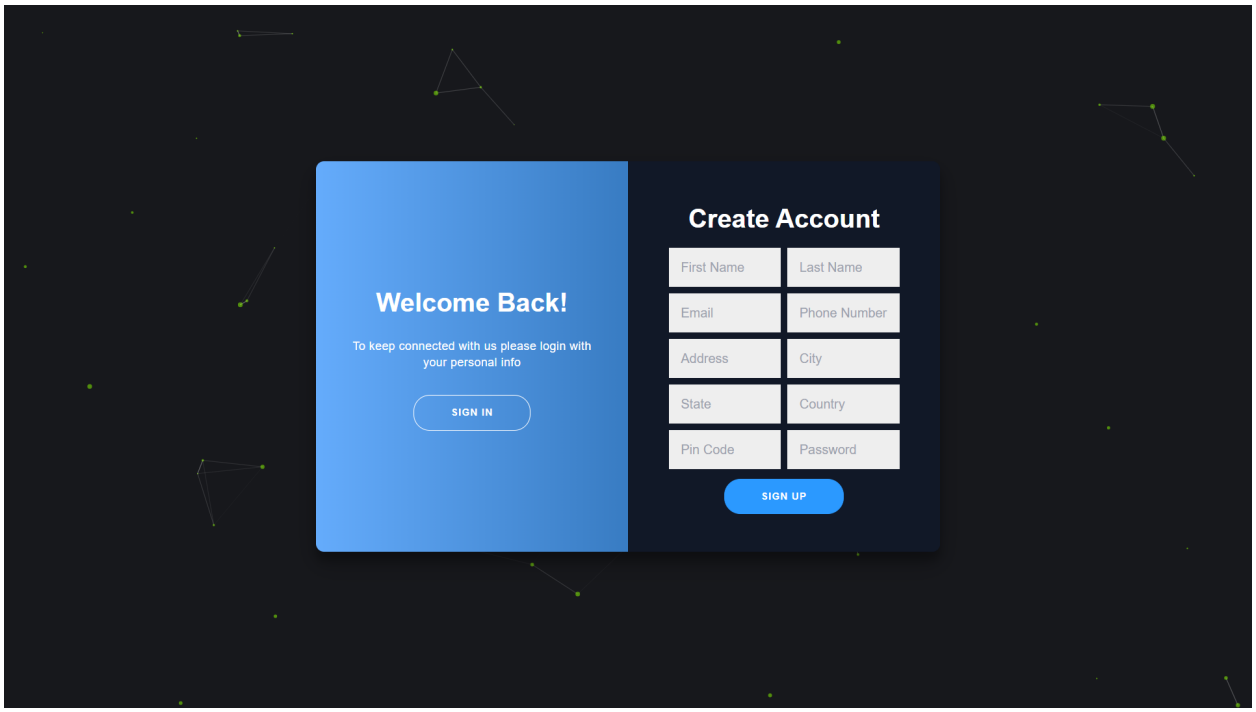
- **UI/UX Test (User Interface):**
  Test Case 7: Ensure that the user interface elements are consistently displayed correctly on different devices and screen sizes.
  Test Case 8: Verify that the user interface follows a consistent design and provides an intuitive user experience.

- **Security Test (Data Protection):**
  Test Case 9: Validate that user data is properly encrypted and protected during transmission.
  Test Case 10: Attempt to access the system without proper authentication to ensure robust denial of unauthorized access.

These test cases are indicative of our comprehensive testing suite, which evaluates various aspects of the "Stalk Your Stock" platform, including functionality, security, and user experience. By adopting this rigorous testing framework, we ensure that the platform provides a reliable and secure environment for users to practice stock trading and gain insights into the financial markets.

Screenshots





HOME PROFILE PORTFOLIO TRANSACTION STOCK LOGOUT

Welcome to Stalk Your Stock

Learn, Invest, and Grow Your Portfolio

Get Started

Learn and Practice

Use our simulator to learn about stock market trading and practice with virtual money. Access a wide range of educational resources, articles, and video tutorials.

Invest Wisely

Make informed investment decisions, explore different stocks, and track your portfolio's performance. Stay updated with real-time stock data, news, and market trends.

Featured Articles

How to Diversify Your Investment Portfolio

Learn the importance of diversification and strategies to spread your investments effectively.

Understanding Stock Market Volatility

Discover the factors contributing to stock market volatility and how to navigate through turbulent times.

Top Stock Picks for 2023

Explore the top stocks that experts are recommending for this year and make informed investment choices.

HOME PROFILE PORTFOLIO TRANSACTION STOCK LOGOUT

Neer Patel

Address: 221B, Baker's street

Email: neer1@gmail.com

Phone No: 07016421539

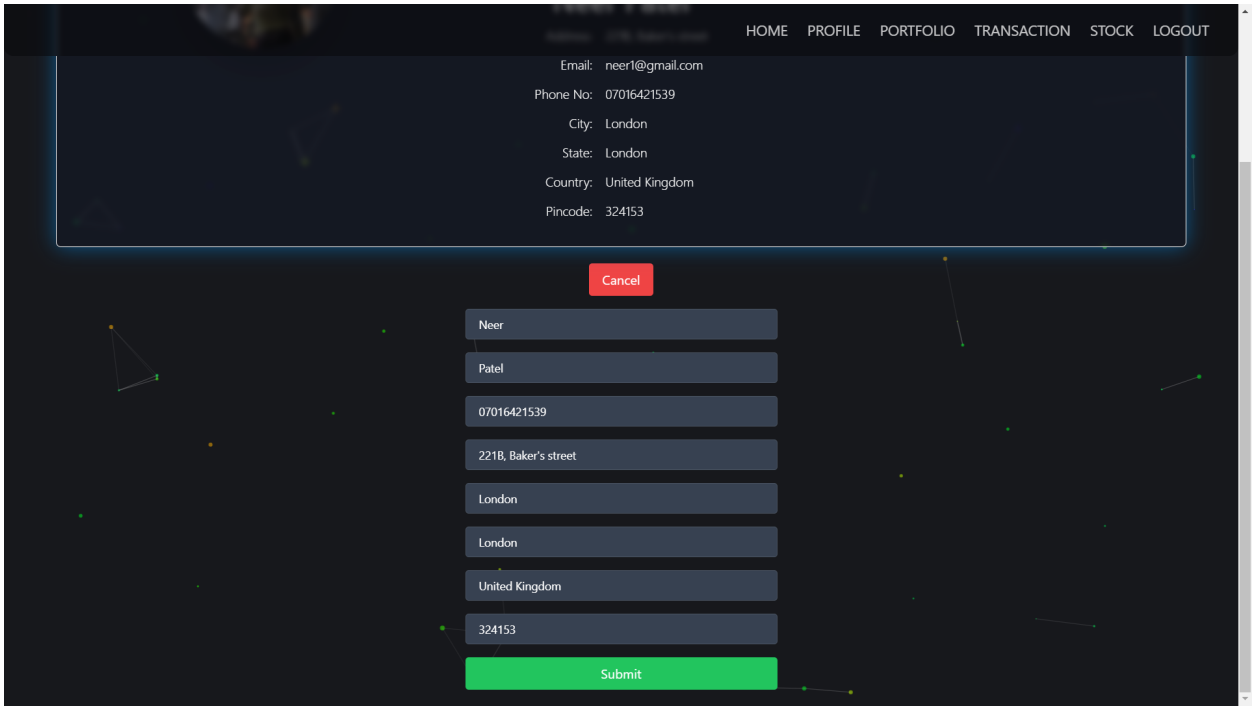
City: London

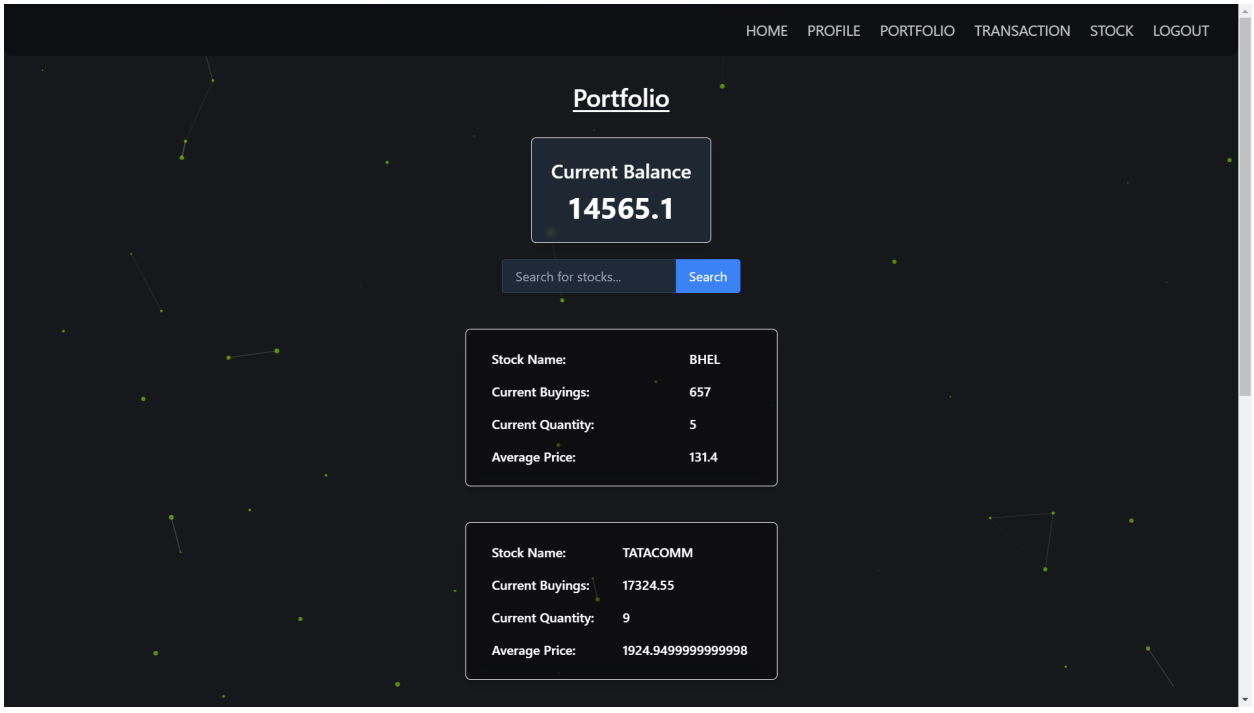
State: London

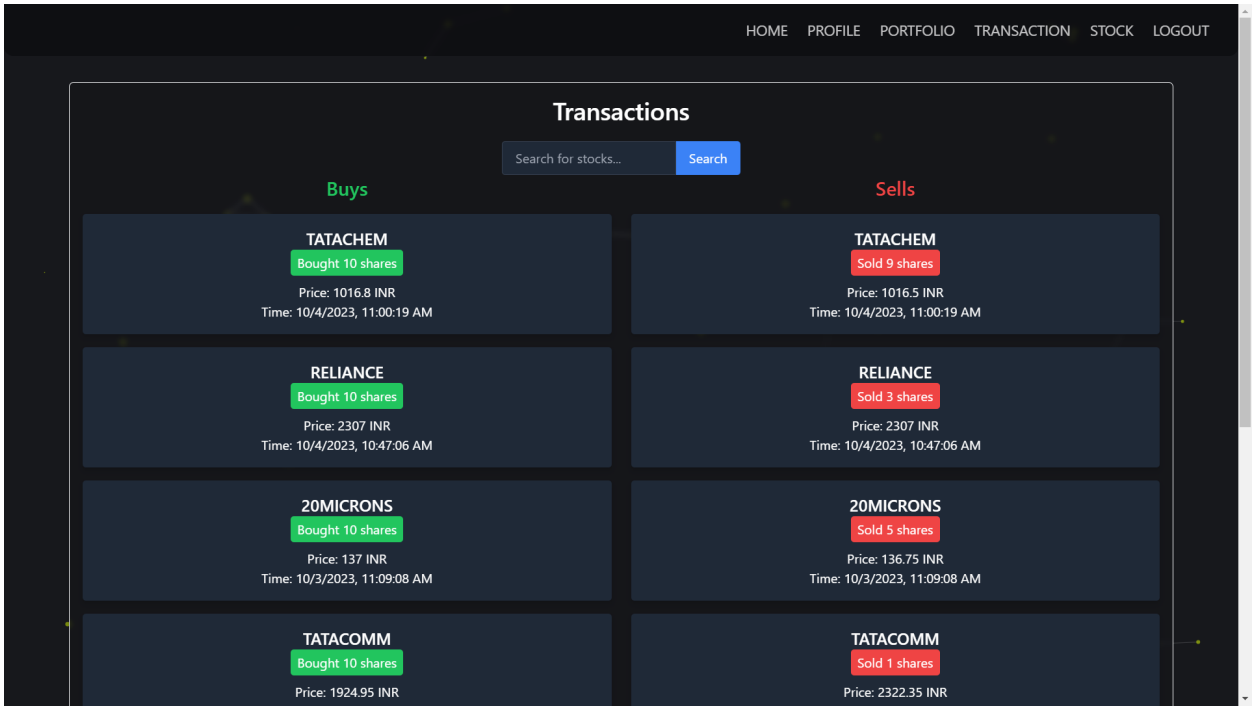
Country: United Kingdom

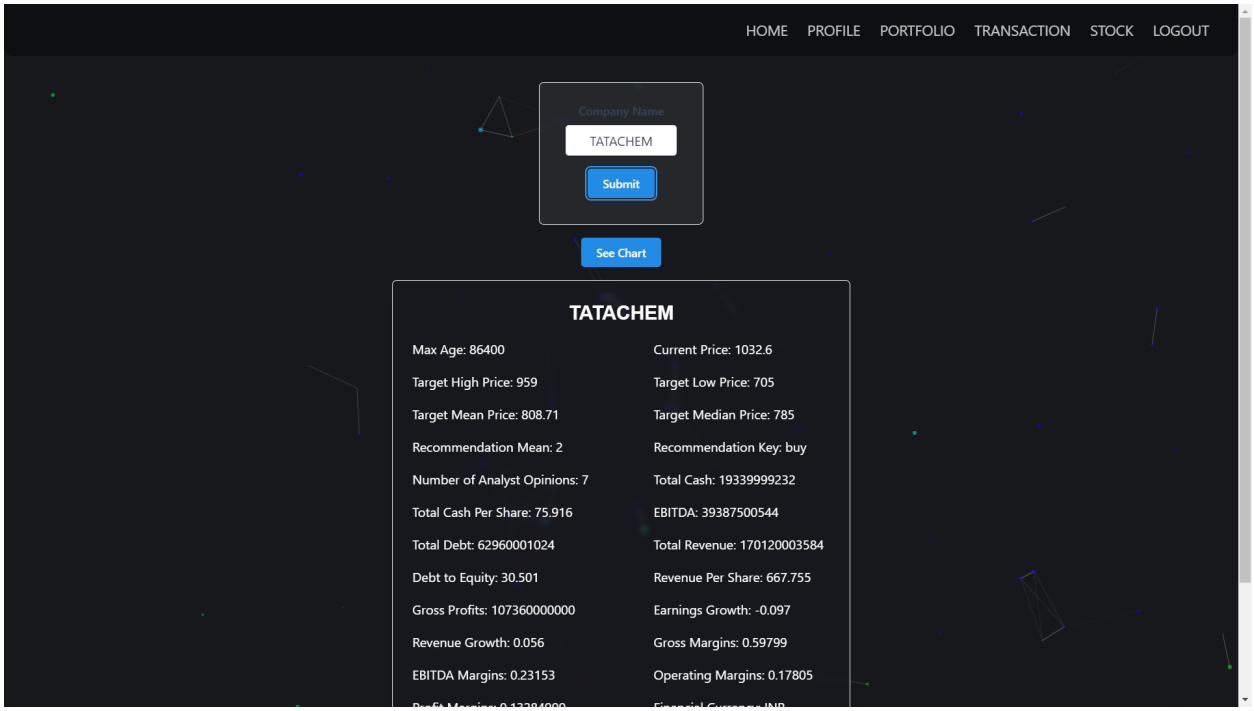
Pincode: 324153

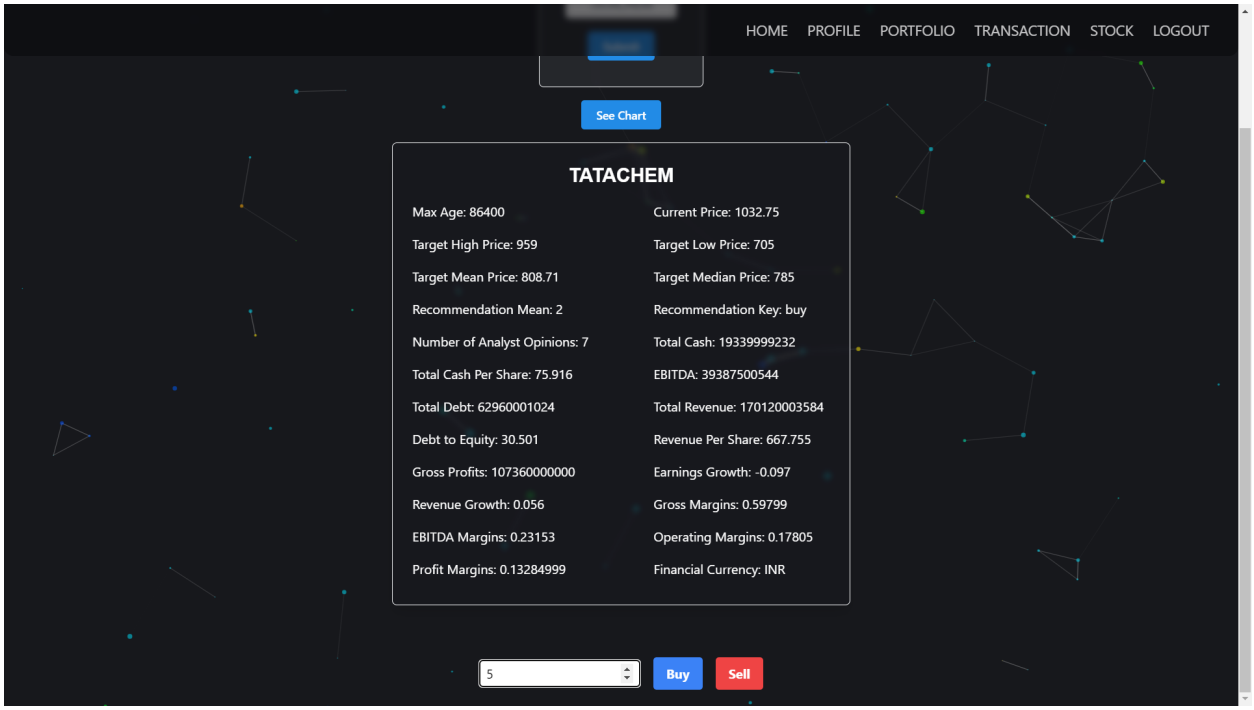
Update Profile

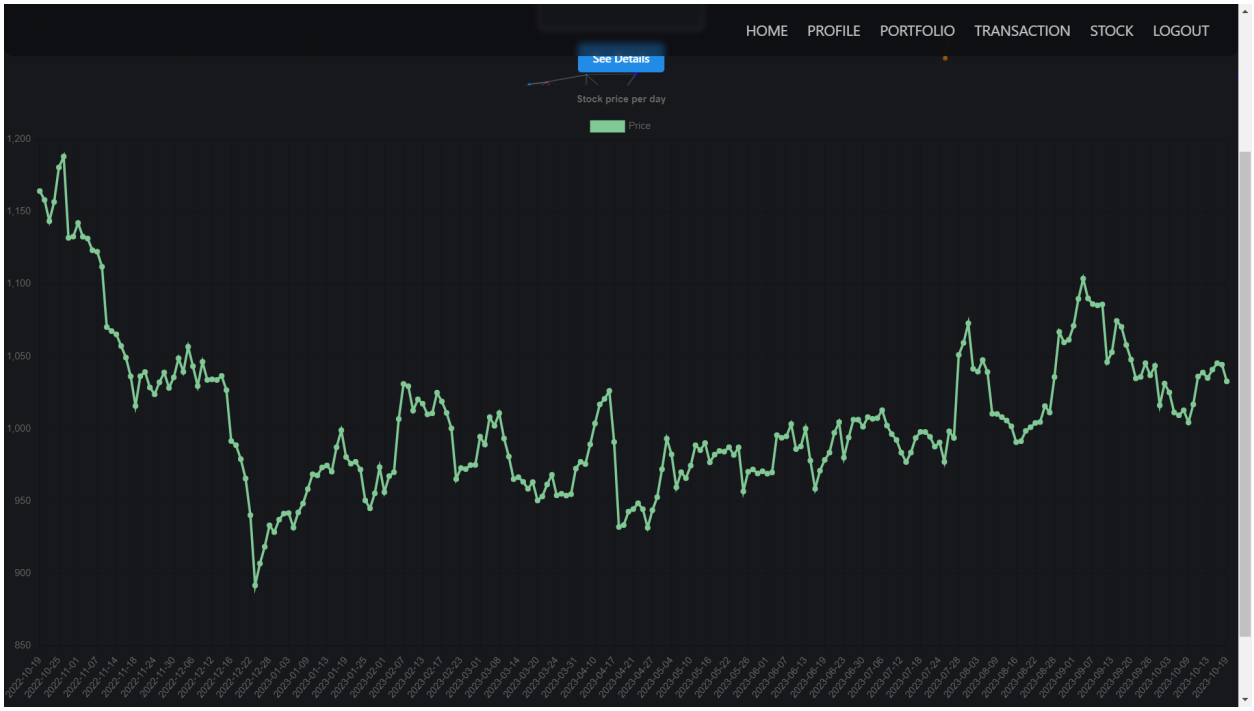

HOME
PROFILE
PORTFOLIO
TRANSACTION
STOCK
LOGOUT
Email: neer1@gmail.com
Phone No: 07016421539
City: London
State: London
Country: United Kingdom
Pincode: 324153
Cancel
Neer
Patel
07016421539
221B, Baker's street
London
London
United Kingdom
324153
Submit


HOME
PROFILE
PORTFOLIO
TRANSACTION
STOCK
LOGOUT
Portfolio
Current Balance
14565.1
Search for stocks...
Search
Stock Name: BHEL
Current Buyings: 657
Current Quantity: 5
Average Price: 131.4
Stock Name: TATACOMM
Current Buyings: 17324.55
Current Quantity: 9
Average Price: 1924.9499999999998


HOME PROFILE PORTFOLIO TRANSACTION STOCK LOGOUT
Transactions
Search for stocks...
Search
Buys
Sells
TATACHEM
Bought 10 shares
Price: 1016.8 INR
Time: 10/4/2023, 11:00:19 AM
TATACHEM
Sold 9 shares
Price: 1016.5 INR
Time: 10/4/2023, 11:00:19 AM
RELIANCE
Bought 10 shares
Price: 2307 INR
Time: 10/4/2023, 10:47:06 AM
RELIANCE
Sold 3 shares
Price: 2307 INR
Time: 10/4/2023, 10:47:06 AM
20MICRONS
Bought 10 shares
Price: 137 INR
Time: 10/3/2023, 11:09:08 AM
20MICRONS
Sold 5 shares
Price: 136.75 INR
Time: 10/3/2023, 11:09:08 AM
TATACOMM
Bought 10 shares
Price: 1924.95 INR
TATACOMM
Sold 1 shares
Price: 2322.35 INR


HOME PROFILE PORTFOLIO TRANSACTION STOCK LOGOUT
Company Name
TATACHEM
Submit
See Chart
TATACHEM
Max Age: 86400
Current Price: 1032.6
Target High Price: 959
Target Low Price: 705
Target Mean Price: 808.71
Target Median Price: 785
Recommendation Mean: 2
Recommendation Key: buy
Number of Analyst Opinions: 7
Total Cash: 19339999232
Total Cash Per Share: 75.916
EBITDA: 39387500544
Total Debt: 62960001024
Total Revenue: 170120003584
Debt to Equity: 30.501
Revenue Per Share: 667.755
Gross Profits: 107360000000
Earnings Growth: -0.097
Revenue Growth: 0.056
Gross Margins: 0.59799
EBITDA Margins: 0.23153
Operating Margins: 0.17805


HOME
PROFILE
PORTFOLIO
TRANSACTION
STOCK
LOGOUT
See Chart
TATACHEM
Max Age: 86400
Current Price: 1032.75
Target High Price: 959
Target Low Price: 705
Target Mean Price: 808.71
Target Median Price: 785
Recommendation Mean: 2
Recommendation Key: buy
Number of Analyst Opinions: 7
Total Cash: 19339999232
Total Cash Per Share: 75.916
EBITDA: 39387500544
Total Debt: 62960001024
Total Revenue: 170120003584
Debt to Equity: 30.501
Revenue Per Share: 667.755
Gross Profits: 107360000000
Earnings Growth: -0.097
Revenue Growth: 0.056
Gross Margins: 0.59799
EBITDA Margins: 0.23153
Operating Margins: 0.17805
Profit Margins: 0.13284999
Financial Currency: INR
5
Buy
Sell


HOME
PROFILE
PORTFOLIO
TRANSACTION
STOCK
LOGOUT
See Details
Stock price per day
Price
1,200
1,150
1,100
1,050
1,000
950
900
850

Conclusion

In conclusion, "Stalk Your Stock" is a web-based stock market simulator that provides users with essential tools to simulate trading and gain insights into stock market dynamics. The project successfully implemented user registration, secure login, and profile management features, enabling users to create accounts, access their profiles, and update personal information. The system effectively simulates stock trading, allocates virtual credits, and maintains transaction histories for users. It also provides real-time stock data through the Yahoo Finance API, ensuring that users have access to the latest market information. Users can search for stocks, analyze their performance, manage portfolios, and track stock prices using interactive charts. The system prioritizes performance, security, usability, and data integrity. "Stalk Your Stock" offers a practical and educational platform for users to refine their trading skills, all within a secure and user-friendly environment.

Limitations and Future Extension

- **Limitations:**

  1. Lack of Real Trading: The project is a stock market simulator and does not support real trading. Users can't invest actual money, limiting the practical experience of real financial risk.

  2. Limited Data Sources: "Stalk Your Stock" relies solely on the Yahoo Finance API for real-time data. The system's accuracy and coverage depend on the reliability and comprehensiveness of this single data source.

  3. Minimal User Guidance: While the platform assumes users possess basic stock trading knowledge, it lacks comprehensive educational materials or guidance for novices.

  4. Data Delay: The real-time data may have a slight delay, which may not accurately reflect actual stock market conditions.

- **Functionalities Not Implemented:**

  1. Advanced Analytics: The system does not offer advanced technical or fundamental analysis tools, which could be beneficial for users looking to perform in-depth stock analysis.

  2. Social Features: The project does not incorporate social networking features, such as discussion forums or collaborative trading.

- **Future Extensions:**

  1. Integration with Multiple Data Sources: In the future, the project could expand to integrate data from multiple sources to enhance data coverage and accuracy.

  2. Educational Resources: Adding educational resources, such as tutorials, articles, or video content, would make the platform more beginner-friendly.

  3. Virtual Competitions: Creating virtual trading competitions or challenges can enhance user engagement and provide a competitive aspect to the platform.

  4. Market News Integration: Integrating real-time market news and alerts to help users stay informed of market developments.

Bibliography

- Stackoverflow is used majorly for error solving.
- TailwindCSS is used for making the web app UI.

  Available: https://tailwindui.com/
- MongoDB Documentation. [Online].

  Available: https://docs.mongodb.com/
- Node.js Documentation. [Online].

  Available: https://nodejs.org/en/docs/
- Express.js Documentation. [Online].

  Available: https://expressjs.com/
- Next.js Documentation. [Online].

  Available: https://nextjs.org/docs/