Intuitive stock trading platform

|  |
| --- |
| **CONTENT** |
| **Abstract** |
| **1.INTRODUCTION** |
| 1.1 Problem Statement |
| 1.2 Objective of the Project |
| 1.3 Scope |
| 1.4 Motivation |
| 1.5 Project Introduction |
| **2.LITERATURE SURVEY** |
| 2.1 Related Work |
| **3. SYSTEM ANALYSIS** |
| 3.1 Existing System |
| 3.2 Disadvantages |
| 3.3 Proposed System |
| 3.4 Advantages |
| 3.5 work Flow of Proposed system |
| **4. REQUIREMENT ANALYSIS** |
| 4.1Function and nonfunctional requirements |
| 4.2 Hardware Requirements |
| 4.3 Software Requirements |
| 4.4 Architecture |
| **5. SYSTEM DESIGN** |
| 5.1 Introduction of Input design |
| 5.2 UML Diagram(class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram) |
| 5.3 Data Flow Diagram |
| **6. IMPLEMENTATION AND RESULTS** |
| 6.1 Modules |
| 6.2 Output Screens |
| 1. **SYSTEM STUDY AND TESTING** |
| 7.1 Feasibility study |
| 7.2 Types of test & Test Cases |
| **8. CONCLUSION** |
| **9. FUTURE ENHANCEMENT** |
| **10. REFERENCES** |

**ABSTRACT**

SB Stocks is a user-friendly, MERN-based stock trading platform designed to elevate the stock trading experience through paper trading. It allows users to access and analyze a wide range of stocks listed on the US Stock market, providing real-time data, historical trends, and comprehensive market insights. The platform's intuitive interface enables users to simulate trading with virtual funds, practice various investment strategies, and monitor their portfolio's performance—all without financial risks. With features like real-time updates, educational resources, and a supportive community, SB Stocks helps both novice and experienced traders refine their skills and gain valuable insights into the world of stock trading.SB Stocks is a user-friendly, MERN-based stock trading platform designed to elevate the stock trading experience through paper trading. It allows users to access and analyze a wide range of stocks listed on the US Stock market, providing real-time data, historical trends, and comprehensive market insights. The platform's intuitive interface enables users to simulate trading with virtual funds, practice various investment strategies, and monitor their portfolio's performance—all without financial risks. With features like real-time updates, educational resources, and a supportive community, SB Stocks helps both novice and experienced traders refine their skills and gain valuable insights into the world of stock trading.

## **PROBLEM STATEMENT**

## In today's dynamic financial landscape, individual investors face significant challenges when attempting to navigate the complex world of stock trading. Many struggle with limited access to real-time market data, lack of user-friendly analysis tools, and difficulty in making informed investment decisions. Traditional stock trading platforms often overwhelm users with excessive information, while failing to provide personalized insights tailored to individual investment goals and risk tolerances. Additionally, the absence of educational resources and community support leaves many novice investors feeling isolated and uncertain.

## SB Stocks aims to address these pain points by developing a comprehensive stock trading and analysis platform using the MERN stack. This innovative solution will democratize access to financial markets, offering users a seamless, intuitive interface that combines real-time market data, advanced analytics, and personalized investment recommendations. By integrating educational resources, social features, and AI-driven insights, SB Stocks will empower both novice and experienced investors to make more informed decisions, fostering a community of knowledgeable and confident traders in an increasingly complex financial ecosystem.

## **MOTIVATION:**

## **OBJECTIVE OF THE PROJECT:**

The objective of the SB Stocks platform is to create a comprehensive and user-friendly environment for stock trading simulation. Key goals include enabling seamless user registration and secure sign-in, providing access to a wide range of stocks listed on the US Stock market, and facilitating the creation and management of virtual portfolios through paper trading. The platform aims to deliver real-time market data, historical stock analysis, and educational resources to empower users in refining their trading strategies without financial risks. Ultimately, SB Stocks strives to enhance the accessibility, learning experience, and effectiveness of stock trading for users of all levels.**SCOPE:**

**SCOPE:**

The scope of **RideEase** covers the development of a comprehensive cab booking platform that caters to users across various domains, including daily commuters, business travelers, and tourists. The project includes features such as user registration, secure login, real-time cab tracking, and flexible ride scheduling. It aims to support multiple ride options, allowing users to choose vehicles based on their preferences while providing accurate fare estimates and arrival times. Additionally, the app will implement secure payment processing and data protection measures, ensuring a reliable and seamless transportation experience for all users.

**PROJECT INTRODUCTION**

In today's dynamic financial landscape, having access to a reliable and user-friendly trading platform is essential for both beginners and experienced traders. The SB Stocks platform, developed using the MERN stack (MongoDB, Express.js, React, Node.js), addresses this need by offering a seamless and intuitive environment for stock trading simulations. Users can easily register and log in to access their personalized trading dashboard, allowing them to explore a wide range of stocks listed on the US Stock market. With the ability to create virtual portfolios and practice trading through paper trading, users can refine their strategies without the risks associated with real-world investments. SB Stocks emphasizes a secure and smooth user experience, featuring user profile management, real-time market data, and educational resources. The platform is designed to empower users to enhance their trading skills, explore new investment strategies, and gain valuable market insights, all in a risk-free and engaging digital space. **2. LITERATURE SURVEY**

2.1 Related Work

 Author(s): S. Chen et al.

Title: "A Comprehensive Review of Stock Trading Platforms"

Outcome: This study provides an extensive overview of various stock trading platforms, their features, and technological foundations, offering insights into their applications for different investor profiles.

Disadvantages: The review lacks in-depth analysis of user experience and educational components, which are crucial for engaging and retaining novice investors.

 Author(s): M. Kumar et al.

Title: "Performance Evaluation of AI-driven Stock Prediction Models"

Outcome: This research analyzes the accuracy and reliability of various AI-driven stock prediction models, assessing their potential for integration into trading platforms.

Disadvantages: The study focuses primarily on technical aspects, overlooking the importance of presenting complex predictions in a user-friendly manner for non-expert investors.

 Author(s): L. Wang et al.

Title: "Enhancing User Engagement in Financial Trading Applications"

Outcome: This paper highlights strategies for improving user engagement in financial trading apps, suggesting features like gamification and social trading to increase user retention.

Disadvantages: The research is limited in addressing the specific needs of long-term investors, focusing more on short-term trading and speculative activities.

**Top of Form**

**Bottom of Form**

**3. SYSTEM ANALYSIS**

**Existing System:**

Current stock trading platforms often present challenges such as steep learning curves, high financial risks, and limited opportunities for beginners to practice trading skills. Many platforms require users to open live accounts and invest real money from the outset, making it difficult for novices to gain experience without incurring potential losses. Additionally, existing applications may lack intuitive interfaces and user-friendly features for managing portfolios, making it challenging for users to analyze stock performance and track their investments efficiently.

**Disadvantages:**

**1.Complex Learning Curve2.Lack of Real-Money Transition3.Limited Real-Time Data4.Dependence on Virtual Funds**

### **Proposed System**

### The proposed SB Stocks platform aims to offer a user-friendly environment that simplifies the stock trading simulation experience. With a streamlined registration and secure login process, users can easily access their trading dashboard and explore a wide range of stocks listed on the US Stock market. The platform enables users to create virtual portfolios, practice trading with virtual funds, and track their investment performance in real-time, providing a risk-free way to refine their trading strategies. Comprehensive stock data, including real-time prices and historical trends, empowers users to make informed decisions. Enhanced security measures ensure that user data remains protected, allowing for a safe and engaging learning experience. SB Stocks addresses the challenges of existing trading platforms by offering a seamless, educational, and accessible solution for both novice and experienced traders.

### **Advantages:**

1. Risk-Free Learning Environment
2. Portfolio Management
3. User-Friendly Interface
4. Secure Account Management
5. Continuous Development

**Work Flow of Proposed system:**

****

**4.REQUIREMENT ANALYSIS**

**4.1 Functional Requirements for SB Stocks**

 User Registration: Users should be able to create an account by providing necessary details such as name, email, and password.

 User Authentication: The system must allow users to securely log in and log out of their accounts.

 Stock Search and Information: Users should be able to search for stocks and view detailed information including real-time prices, historical data, and company profiles.

 Portfolio Management: Users must be able to create and manage multiple investment portfolios, adding or removing stocks as needed.

 Real-Time Market Data: The application should provide real-time market data, including stock prices, indices, and relevant news.

 Trade Execution: Users should be able to place buy and sell orders for stocks, with options for market, limit, and stop orders.

 Watchlist Creation: Users should be able to create and manage watchlists of stocks they're interested in monitoring.

 Analytics and Reporting: The system should provide analytical tools and generate reports on portfolio performance, including gains/losses and dividend income.

 Notification System: Users should receive notifications for price alerts, order executions, and important market news.

 Educational Resources: The application should offer educational content, including tutorials, articles, and webinars on investing and trading strategies.

**4.2 Non-Functional Requirements for SB Stocks**

 Performance: The app should handle multiple simultaneous user requests and data updates without significant latency.

 Scalability: The system must be able to scale to accommodate a growing number of users and increasing market data volume.

 Reliability: The application should ensure high availability, minimizing downtime especially during market hours.

 Security: User data, financial information, and transactions should be securely encrypted, ensuring privacy and protection against unauthorized access.

5. Usability: The user interface should be intuitive and easy to navigate, catering to both novice and experienced investors.

 Compatibility: The application must work across various devices (desktops, tablets, smartphones) and browsers, with a responsive design.

 Data Accuracy: The system should ensure the accuracy and timeliness of financial data, with mechanisms to verify and correct any discrepancies.

 Compliance: The application must comply with relevant financial regulations and data protection laws.

 Maintainability: The codebase should be well-organized and documented for easy updates and maintenance, allowing for quick fixes and new feature implementations.

 Backup and Recovery: The system should have robust mechanisms for regular data backups and a comprehensive recovery plan to protect against data loss or system failures.

11. Performance under Load: The system should efficiently manage peak loads, especially during high-volume trading periods or market volatility.

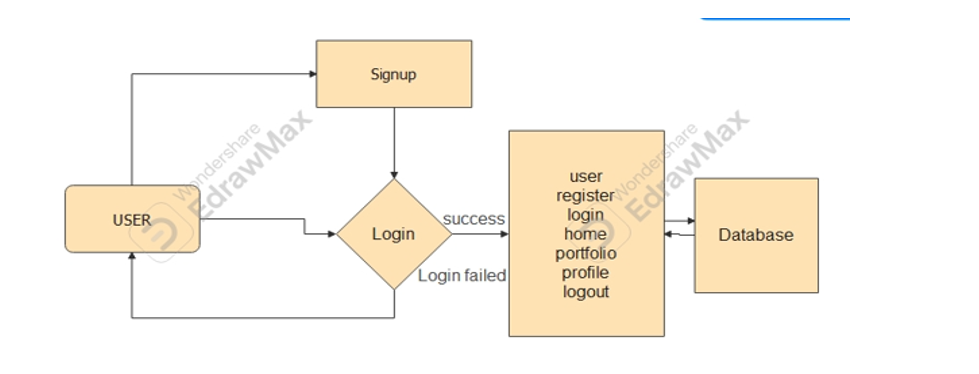
 API Integration: The application should seamlessly integrate with various financial data providers and brokerage APIs for real-time data and trade execution.

1. .

**4.3 SOFTWARE REQUIREMENTS:**

**SOFTWARE SYSTEM CONFIGURATION:**Operating System : Windows 10/11Server side Script : Express jsProgramming Language : java ScriptIDE/Workbench : VS CodeDatabase : Mongo dbClient Side : React Js**HARDWARE REQUIREMENTS:**Processor : I3/Intel ProcessorRAM : 4GB (min)Hard Disk : 160GB

**4.4 Architecture:**



**5. System Design**

**5.1 Introduction of Input Design:**

In an information system like SB Stocks, input refers to the data provided by users that is processed to produce meaningful outputs, such as stock quotes, portfolio summaries, and trade confirmations. Input design for the SB Stocks application should consider the following aspects to ensure high-quality output:

• Input Devices: Users will primarily interact with the system via mobile devices, tablets, or PCs. Keyboards, touchscreens, and potentially voice commands can be used as input devices. For advanced users, integration with external data feeds or APIs might be considered.

Well-designed input forms and screens for SB Stocks should possess the following properties:

• They should serve specific purposes effectively, such as placing trade orders, creating watchlists, or updating user profiles.

• Ensure completion with accuracy, such as preventing incomplete trade orders or inaccurate portfolio allocations.

• Be user-friendly and easy to understand for both novice and experienced investors.

• Focus on user attention, ensuring consistency, clarity, and simplicity to make the investment process straightforward.

By using basic design principles, the input design will address:

• Required Inputs: User data (name, address, financial information), stock data (ticker symbols, order types, quantities), and system inputs (real-time market data, news feeds).

• User Interaction: End users must interact smoothly with input elements like search bars, order forms, and portfolio management tools.

Objectives for Input Design in SB Stocks:

• To design effective data entry and input procedures for users, from basic account management to complex trading strategies.

• To reduce input volume, streamlining the trading process with features like autocomplete for stock symbols and saved order templates.

• To design source documents or other data capture methods to streamline processes like account funding and tax reporting.

• To design input data records and screens, such as user registration, placing trades, and portfolio rebalancing screens.

• To use validation checks and develop effective input controls, ensuring accuracy and completeness in data input, such as preventing invalid trade orders or incomplete account information.

Output Design:

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

Output Design for SB Stocks:

The design of output for the SB Stocks application is critical for ensuring clear communication between the system and its users. Proper output design allows the system to provide users with real-time market data, portfolio performance, and trade confirmations in a timely and understandable manner.

Objectives of Output Design in SB Stocks:

• To develop output design that serves the intended purpose: For SB Stocks, output should focus on delivering key information such as stock quotes, portfolio valuations, and trade execution reports.

• To meet user requirements: The design should cater to both novice investors (e.g., showing simplified portfolio summaries) and experienced traders (e.g., providing detailed technical analysis).

• To deliver the appropriate quantity of output: This could include real-time stock tickers, periodic portfolio reports, and trade notifications, ensuring that only relevant information is presented to the user.

• To format the output appropriately: Outputs like stock charts, financial statements, and performance metrics must be formatted to be easily readable and interpretable, whether on mobile devices or desktop systems.

The objectives of input design are:

• To develop output design that serves the intended purpose and eliminates the production of unwanted output.

• To develop the output design that meets the end user's requirements.

• To deliver the appropriate quantity of output.

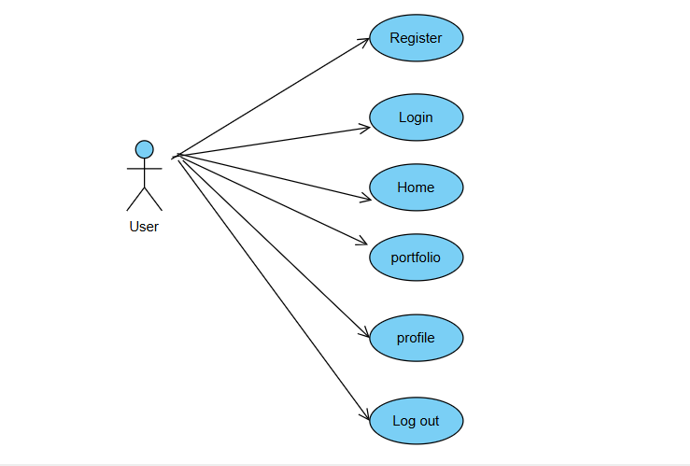
• To form the output in appropriate format and direct it to the right person.

• To make the output available on time for making good investment decisions.

**5.2 UML Diagrams:**

**5.2.1 Use Case Diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Usecase analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



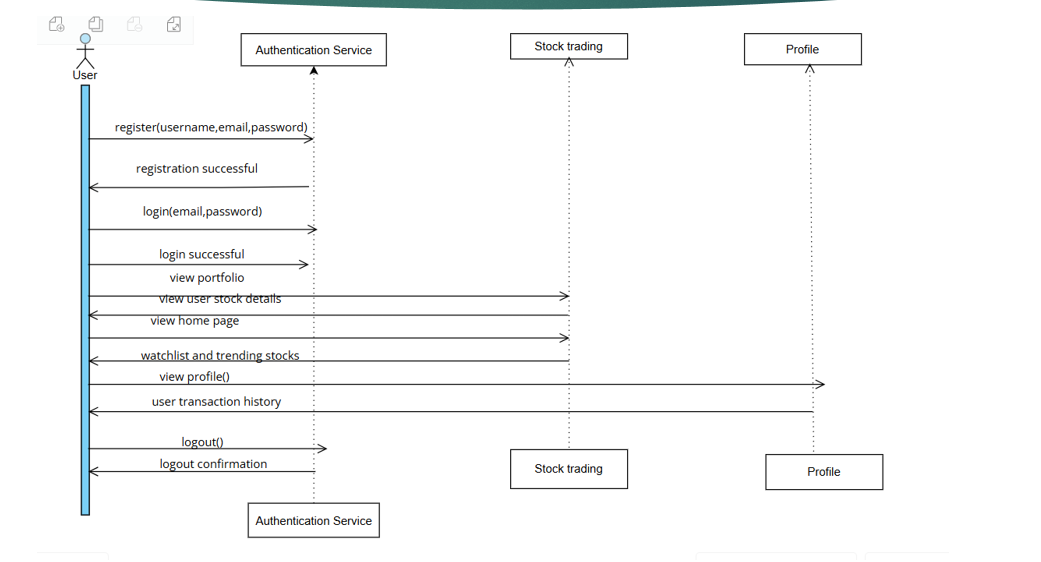
**5.2.2 Class Diagram:**

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



**5.2.3 Sequence Diagram:**

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

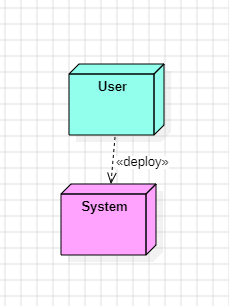


**5.2.4 Collaboration Diagram:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.

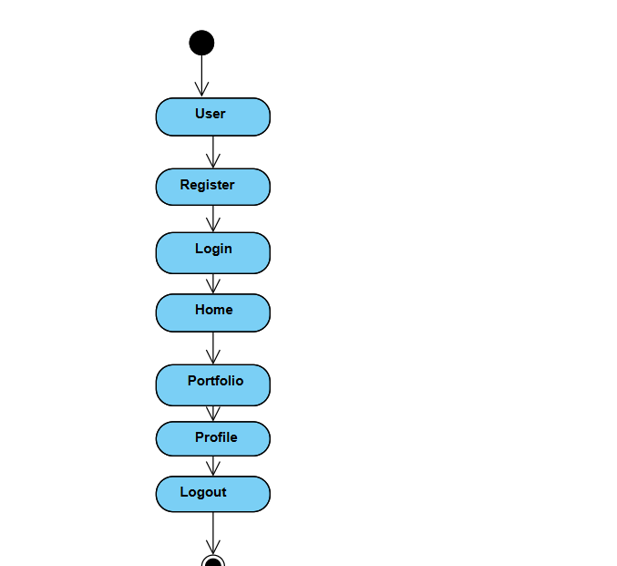
**5.2.5 Deployment Diagram**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



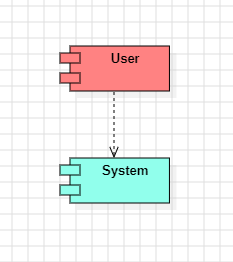
**5.2.6 Activity Diagram:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational stepbystep workflows of components in a system. An activity diagram shows the overall flow of control.



**5.2.7 Component Diagram**:

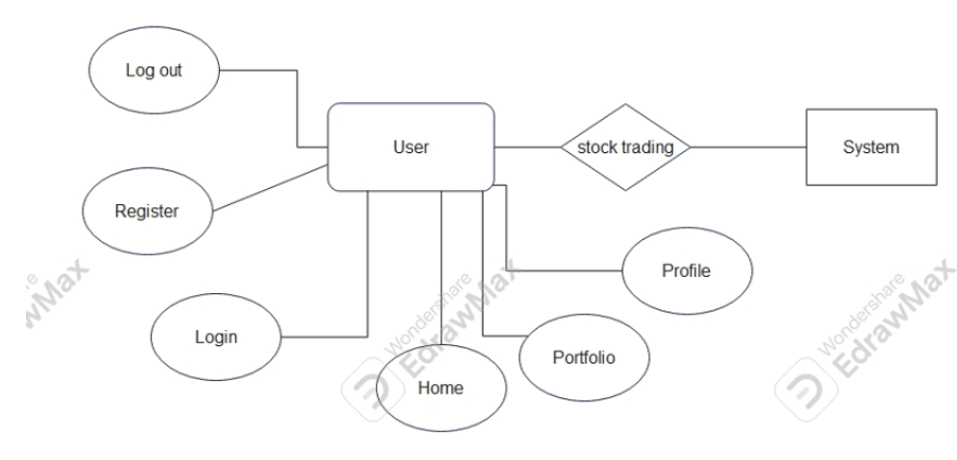
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and doublecheck that every aspect of the system's required functions is covered by planned development.



**5.2.8 ER Diagram:**

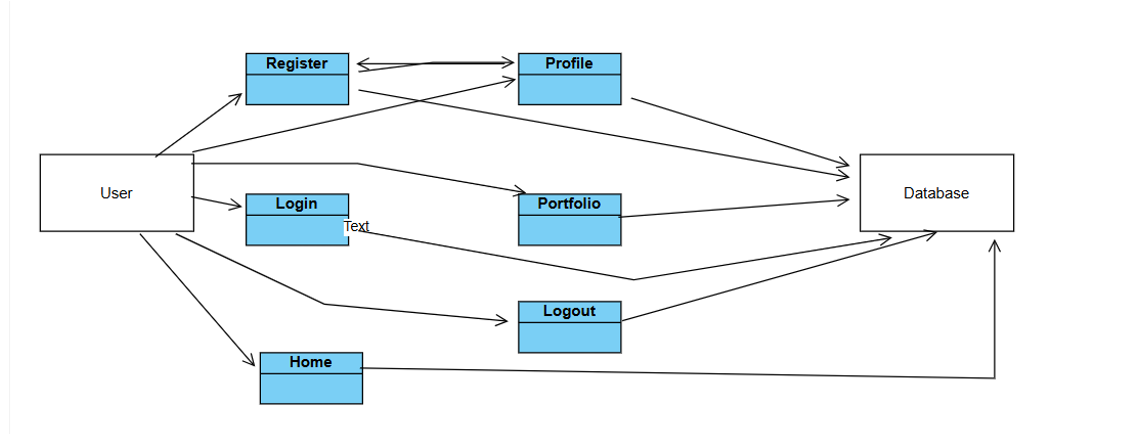
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of ER model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

****

**5.3 DFD Diagram:**

A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

****

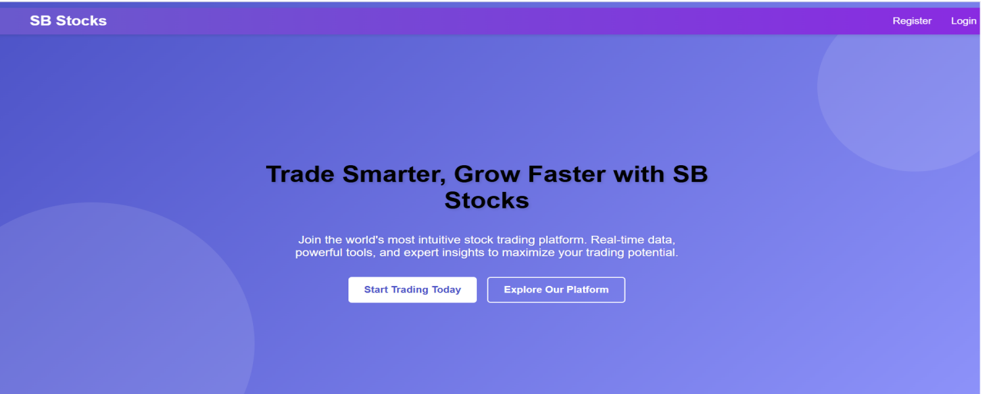
**6. IMPLEMENTATION AND RESULTS**

**6.1. Modules for SB Stock Application:**

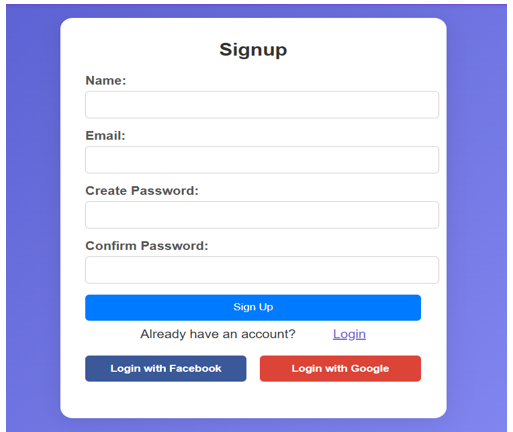
**User Module**Register: User registration with personal details.Login: Secure login for users to access their account.Home: Trending stcoks and watchlist of the user.Portfolio: particular user stock details are displayed.Profile: Transaction history of the particular user and the balance amount will be displayedLogout: Secure logout to end the session.

**6.2 Output Screens:**

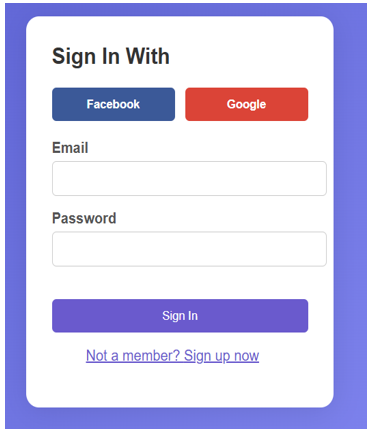
**Homepage:** Default landing page for user where he will get the information about the project

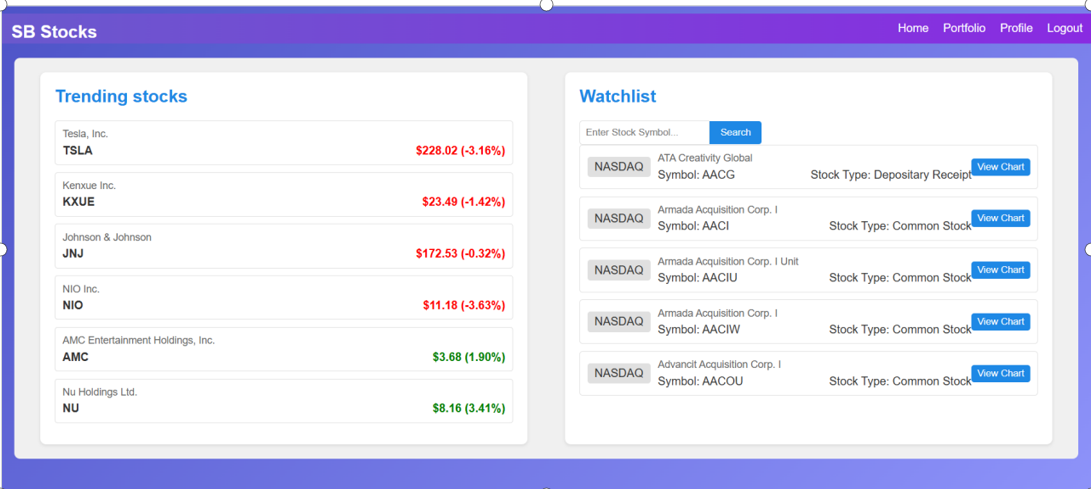


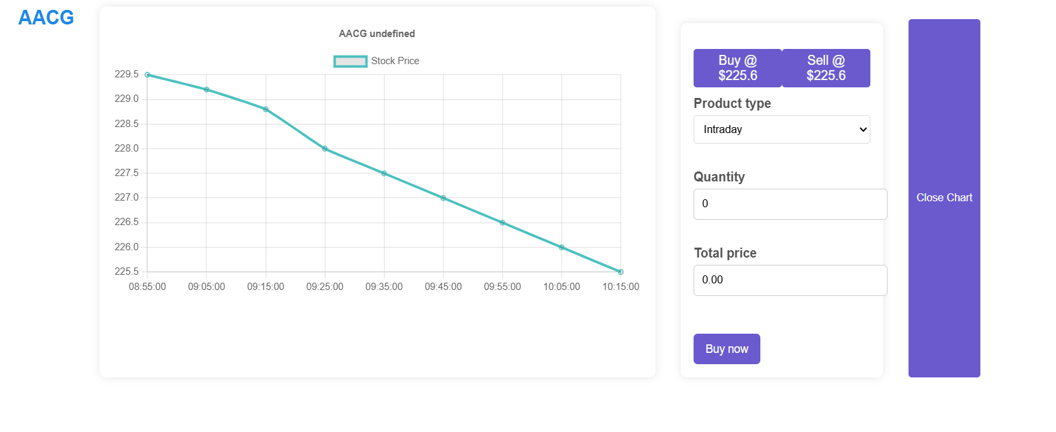
**Figure 2: Register page:** Save user data to the database and send a confirmation email

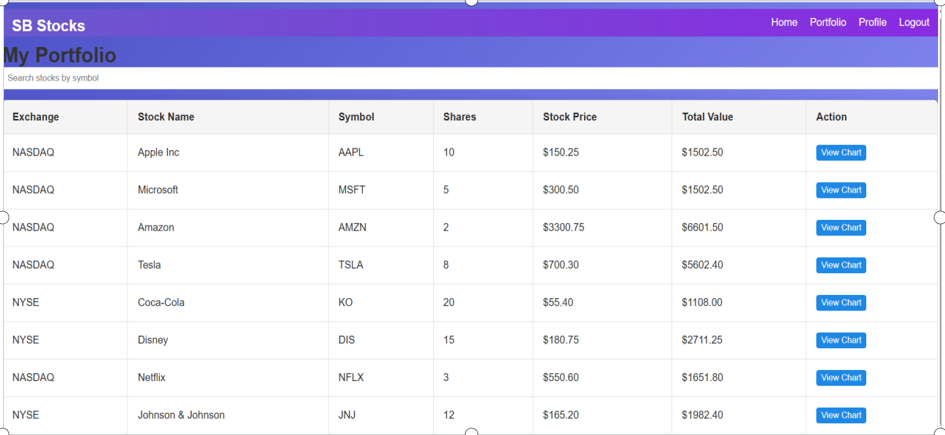
****

**Figure 3: Login page,** Authenticate user and start session**.**

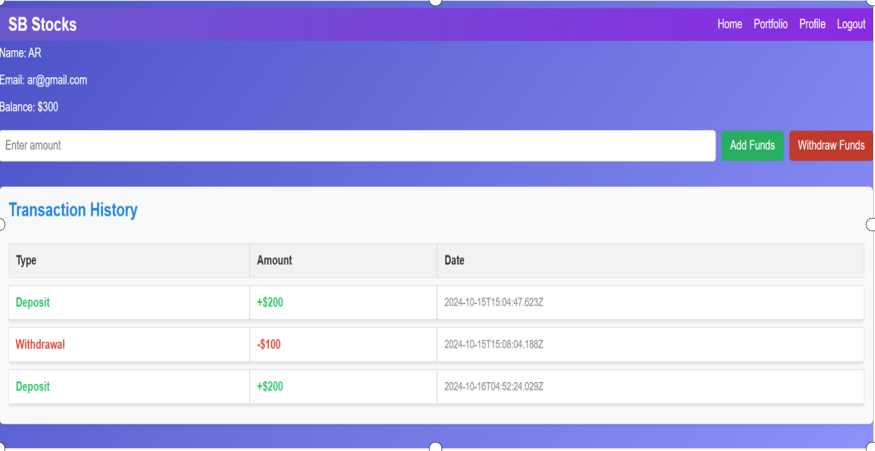
****

****

**Portfolio:** They can view the user transaction related to stock.



**Profile:**  The user can see the transaction history and the balance amount.

****

**7. SYSTEM STUDY AND TESTING**

**7.1 Feasibility Study**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* Economic feasibility
* Technical feasibility
* Social feasibility

**Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.2 Types of Tests**

**7.2.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.2.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.2.3 Functional testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.2.4 White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.2.5 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and rsesponses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page**.**

**3 TEST CASES:**

**User Module:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Precondition** | **Test Steps** | **Expected Result** |
| TC01 | Register a new user | User is on the registration page | 1. Enter valid user details (e.g., username, email, password).  2. Click on "Register" button. | User is successfully registered, and a confirmation message appears. |
| TC02 | Register with existing email | User is on the registration page | 1. Enter details with an existing email.  2. Click on "Register" button. | Registration fails with an error message stating that the email is already in use. |
| TC03 | Login with valid credentials | User is on the login page | 1. Enter correct email and password.  2. Click on "Login" button. | User is successfully logged in and redirected to the dashboard. |
| TC04 | Login with invalid credentials | User is on the login page | 1. Enter incorrect email or password.  2. Click on "Login" button. | Login fails with an error message indicating incorrect email or password. |
| TC08 | View user profile | User is logged in | 1. Navigate to the "Profile" section. | User profile details (e.g., name, email, etc.) are displayed. |
| TC09 | Update profile information | User is logged in and on Profile page | 1. Click on "Edit Profile".  2. Modify profile details.  3. Click on "Save" button. | Profile information is updated, and a success message is shown. |
| TC10 | Logout from the application | User is logged in | 1. Click on the "Logout" button. | User is logged out, and the session is terminated, redirecting to the login page. |

**8 .FUTURE ENHANCEMENT**

As the world of stock trading continues to evolve, several enhancements can be made to further refine and elevate the SB Stocks platform experience. First, integrating real-time market data would provide users with accurate and up-to-the-minute information, making the paper trading experience more reflective of actual market conditions. Additionally, incorporating social trading and leaderboards would enable users to follow and learn from top-performing traders, encouraging community engagement and peer-driven learning.To further enhance the platform's capabilities, AI-powered trading insights could be introduced to offer personalized stock suggestions based on market trends and user preferences. Expanding the app's accessibility by developing a mobile version for iOS and Android would allow users to manage their portfolios and execute trades on the go. **9.CONCLUSION**

SB Stocks is designed to transform your stock trading experience, offering a seamless and user-friendly platform for aspiring and experienced traders alike. Our web app provides easy access to a wide array of stocks from the US Stock market, allowing users to practice their trading skills through paper trading without the financial risks.With real-time stock prices, historical data, and detailed performance analytics, SB Stocks ensures that you can make well-informed trading decisions. Whether you're building virtual portfolios or testing new strategies, our platform delivers a realistic trading environment for refining your skills

**10. REFERANCES**

Real-Time Market Data IntegrationAlpha Vantage API: Offers real-time and historical stock market data. A widely-used service for financial data feeds.  
Website: Alpha VantageYahoo Finance API: Provides access to financial data for stock market tracking and analytics.  
Website: Yahoo Finance APISocial Trading and LeaderboardseToro Platform: Known for pioneering social trading and offering copy trading features.  
Case Study: eToro Social TradingZulutrade: Another platform known for enabling social trading and providing leaderboards.  
Website: ZulutradeAI-Powered Trading InsightsKavout AI: Provides AI-powered stock ranking and insights using advanced algorithms.  
Website: Kavout