

Project Report Format

1. INTRODUCTION

1.1 Project Overview

SmartTrade is a modern, user-friendly web application designed to simplify stock trading for both beginner and experienced investors. The platform provides real-time market data, portfolio management tools, and seamless buying and selling of stocks in a secure and intuitive environment.

The goal of SmartTrade is to offer users an all-in-one solution for managing their stock investments while ensuring an engaging and responsive user experience across all devices.

1.2 Purpose

The purpose of the Stock Trading Web App is to provide users with a robust platform to manage and trade stocks. The app aims to offer real-time data, efficient trading options, and comprehensive portfolio management tools.

2. IDEATION PHASE

2.1 Problem Statement

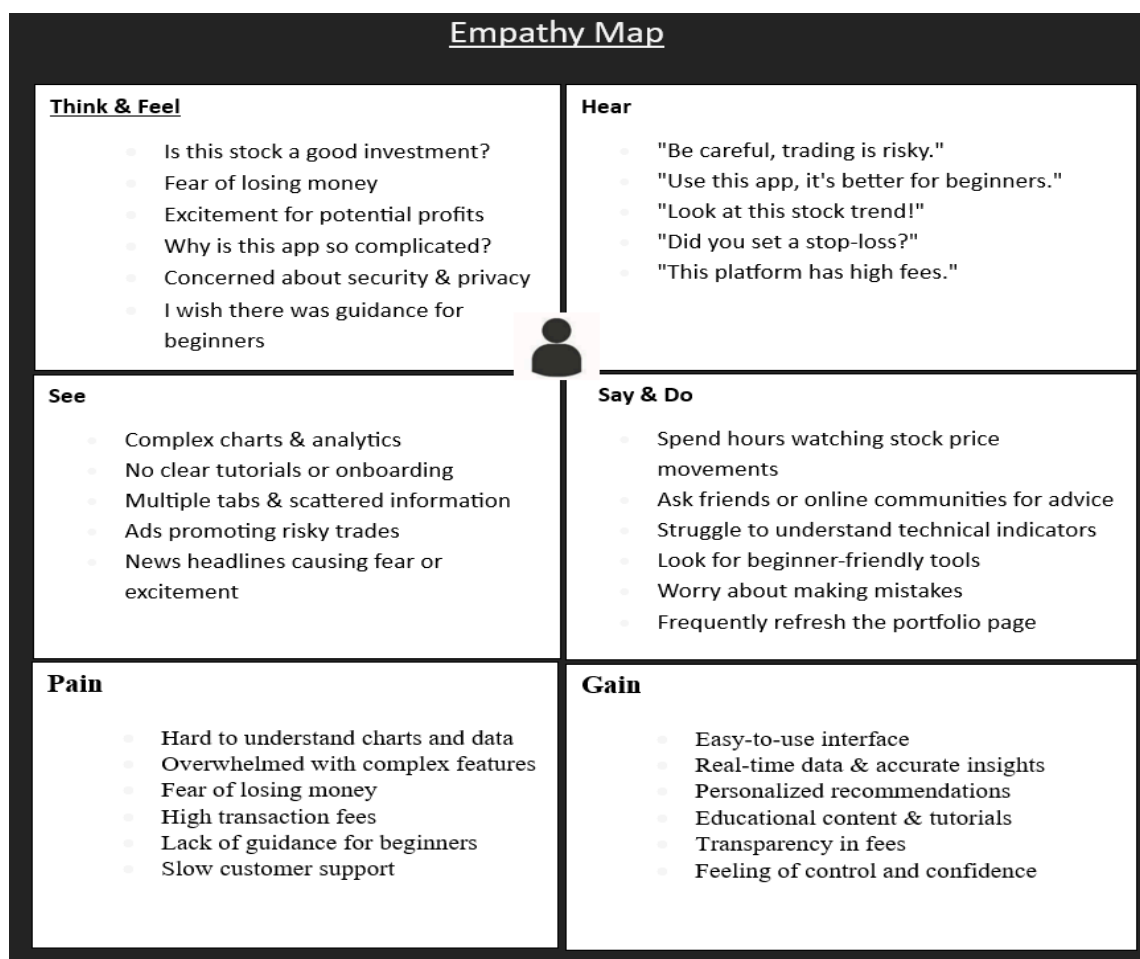
Problem 1: How can we design and implement a robust stock trading web application using the MERN stack that ensures real-time data synchronization across multiple users and devices while maintaining high scalability and performance?

Problem 2: How can we ensure the security, privacy, and regulatory compliance of a stock trading web application built with the MERN stack, while providing a seamless and user-friendly experience?

I am	I'm trying to	But	Because	Which makes me feel
A stock trader	Get real-time updates on stock prices and manage my portfolio efficiently	The application is slow and data is not synchronized in real-time	The system lacks efficient real-time data handling and scalability	Frustrated and unable to make timely trading decisions
A stock trader	Ensure my personal and financial data is secure while trading	I'm concerned about the security measures and regulatory compliance of the application	The application doesn't clearly demonstrate robust security practices and compliance with financial regulations	Anxious and distrustful of the platform

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Rikesh (A stock trader)	Get real-time updates on stock prices and manage my portfolio efficiently	The application is slow and data is not synchronized in real-time	The system lacks efficient real-time data handling and scalability	Frustrated and unable to make timely trading decisions which decreases productivity
PS-2	Rohan (A stock trader)	Ensure my personal and financial data is secure while trading	I'm concerned about the security measures and regulatory compliance of the application	The application doesn't clearly demonstrate robust security practices and compliance with financial regulations	Anxious and distrustful of the platform

2.2 Empathy Map Canvas



2.3 Brainstorming



Brainstorm & idea prioritization

1 Problem Statement

Design and develop a web-based platform for real-time stock trading that allows users to monitor markets, execute trades, manage portfolios, and access educational content. The system should prioritize performance, security, and user experience.

2

Brainstorm

Mitul

What are the possible ways to add a real time data?

What will be our main standout feature?

How will the UI look so that it become more user friendly?

Nupur

How might we implement two-factor authentication in a user-friendly way?

What if users could simulate trades before doing them for real?

What will be the minimum transaction amount to deposit for a user?

Vaasu

How might we reduce data latency to under 1 second for real-time updates?

What all permissions do we need from the user from the device?

How might we make the trade execution process both fast and secure?

Divyanshi

What all permissions do we need to take from NSD and DSE to show the real time data and transaction?

How might we ensure seamless navigation across different parts of the app?

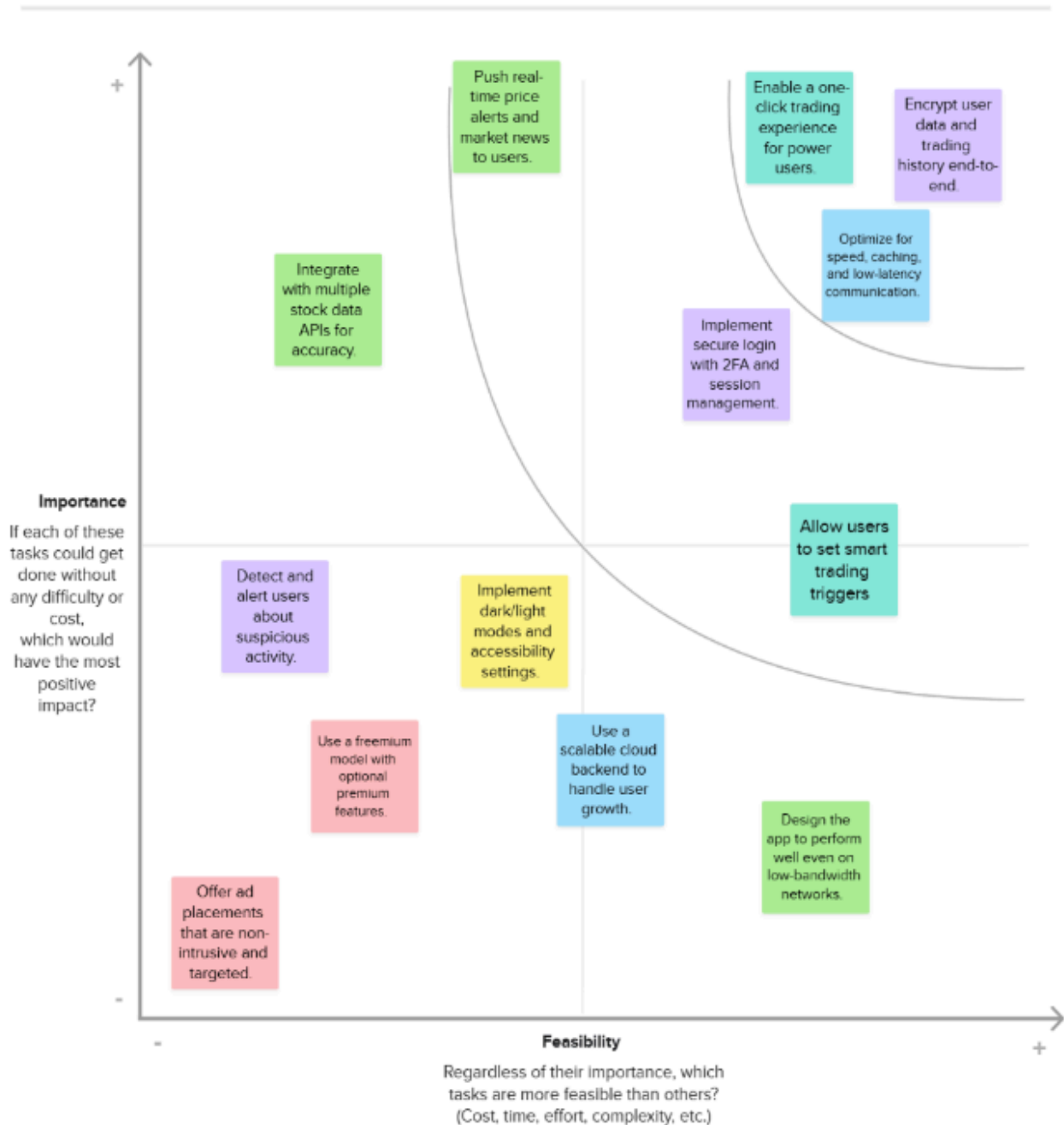
What if users could receive personalized alerts based on stock performance or news?

3

Group ideas

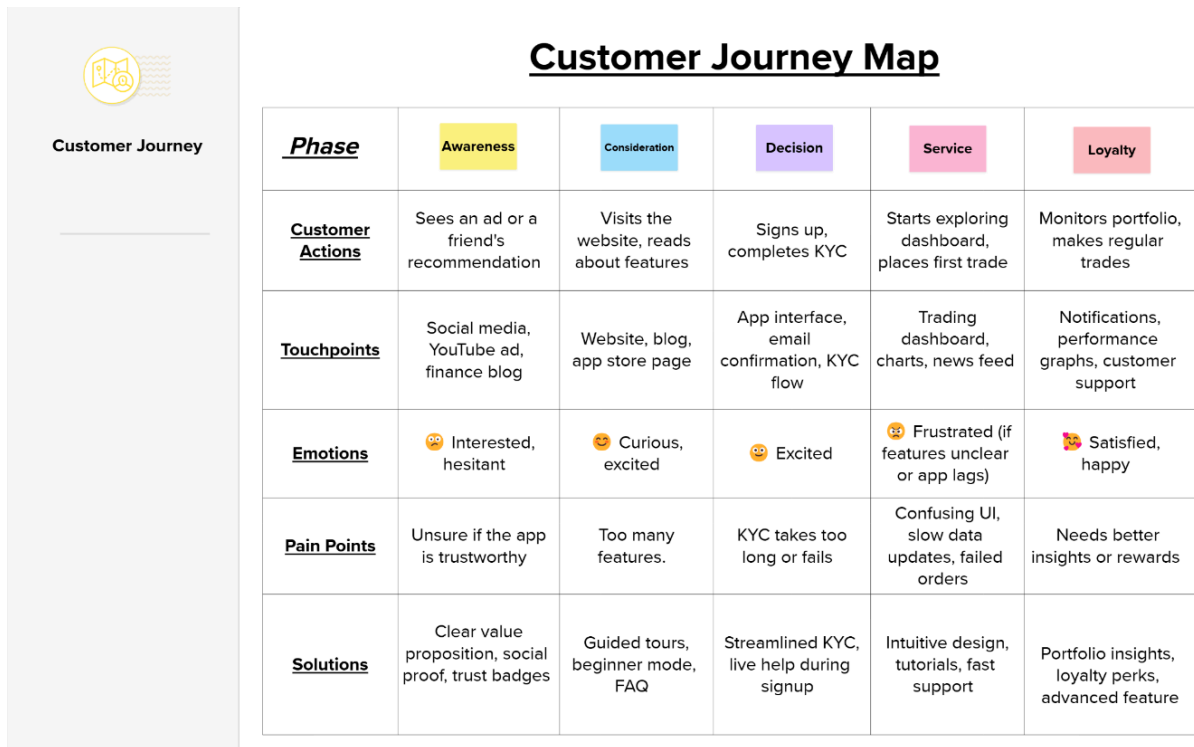


Prioritize



3. REQUIREMENT ANALYSIS

3.1 Customer Journey Map



3.2 Solution Requirement

❖ Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIn
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login via Form Login via Social Media Accounts
FR-4	User Portfolio Management	View Portfolio Edit Portfolio
FR-5	Stock Market Information	Real-time Stock Quotes Historical Data
FR-6	Trading Functionality	Buy Stocks Sell Stocks
FR-7	Alerts and Notifications	Price Alerts News Updates

FR-8	Reporting and Analytics	Performance: The application must load quickly and handle a high number of simultaneous requests. Implement performance optimizations such as caching (e.g., Redis), content delivery networks (CDN), and efficient database queries. Aim for load times under 2 seconds.
------	-------------------------	---

❖ Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	The interface must be intuitive and easy to navigate for users of all levels. This includes clear navigation menus, accessible help resources, and a clean design layout.
NFR-2	Security	Implement robust security measures to protect user data and transactions. This includes encryption (SSL/TLS), secure authentication mechanisms (e.g., OAuth2, JWT), regular security audits, and compliance with industry standards like OWASP.
NFR-3	Reliability	The system should consistently perform as expected without failures. Implement failover ensure data integrity and application stability.
NFR-4	Performance	The application must load quickly and handle a high number of simultaneous requests. Implement performance optimizations such as caching (e.g., Redis), content delivery networks (CDN), and efficient database queries. Aim for load times under 2 seconds.
NFR-5	Availability	Ensure the system is available 99.9% of the time. Use cloud services with multi-region deployments, load balancers, and regular maintenance schedules to minimize downtime.
NFR-6	Scalability	The system should be able to handle increased loads without performance degradation. Implement horizontal scaling for web servers and databases, use microservices architecture where applicable, and utilize cloud infrastructure to dynamically adjust resources based on demand.

3.3 Data Flow Diagram

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Stock Trading Web App DFDs:

Level-0 Description:

User: The end-user who interacts with the stock trading web app.

Process Trade: Handles trade requests from the user, processes them, and updates the stock data and user data.

Fetch Stock Data: Requests and retrieves real-time stock data from the Stock Market Data API and updates the stock data in the database.

Notify User: Sends notifications to the user regarding their trades or stock updates.

Stock Data: Database storing stock information.

User Data: Database storing user information.

Stock Market Data API: External service providing real-time stock market data.

Notification Service: External service used to send notifications to the user.

Level-1 Description and Processes:

User: The end-user who interacts with the stock trading web app.

Stock Data: Database storing stock information.

User Data: Database storing user information.

Stock Market Data API: External service providing real-time stock market data.

Notification Service: External service used to send notifications to the user.

Processes:

User Interface (UI): Manages user requests and interactions with the web app.

User Authentication (Auth): Handles login and signup processes, updates user data.

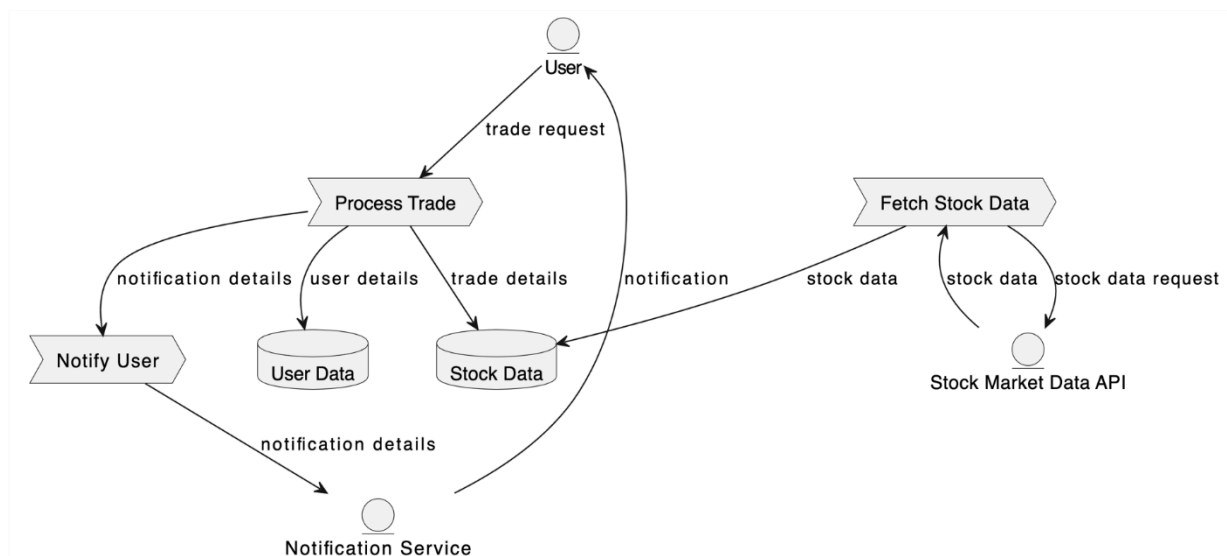
Trade Processor: Manages trade requests, updates stock data, and communicates with the portfolio manager and notifier.

Portfolio Manager: Updates user portfolios based on trade activities.

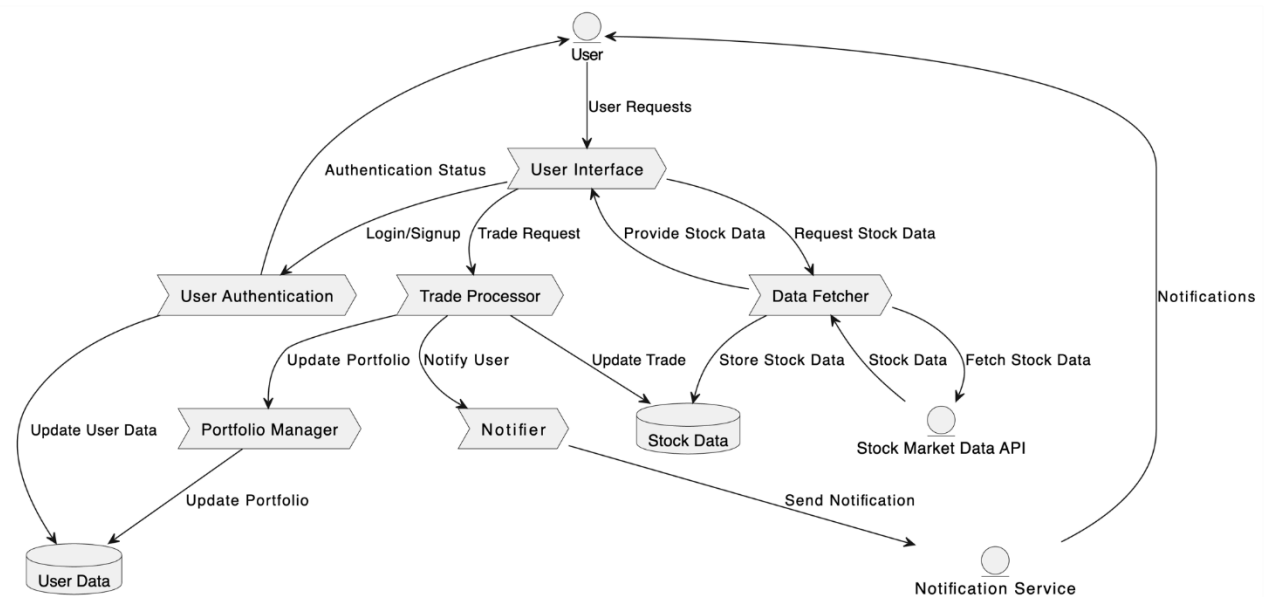
Data Fetcher: Requests real-time stock data from the Stock Market Data API and updates the stock data.

Notifier: Sends notifications to users through the Notification Service.

Level – 0 DFD



Level – 1 DFD



User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release
Customer (Web user)	Account Management	USM-1	As a user, I want to create an account to start trading.	I can provide personal information, set a username and password, and agree to terms of service.	High	Sprint 1
Customer (Web user)	Account Management	USM-2	As a user, I want to log in to my existing account.	I can enter my credentials (username/email and password) and access my account dashboard.	High	Sprint 1
Customer (Web user)	Account Management	USM-3	As a user, I want to securely reset my password.	I can request a password reset link via email and successfully change my password.	Medium	Sprint 2
Customer (Web user)	Market Data	USM-4	As a user, I want to view real-time stock quotes and market data.	I can see up-to-date stock prices, trading volume, and market indices.	High	Sprint 1
Customer (Web user)	Market Data	USM-5	As a user, I want to search for stocks by ticker symbol or company name.	I can quickly find relevant stocks and view their details.	Medium	Sprint 1
Customer (Web user)	Trading	USM-6	As a user, I want to place buy and sell orders for stocks.	I can specify the stock, quantity, price, and order type (market, limit, stop).	High	Sprint 2
Customer (Web user)	Trading	USM-7	As a user, I want to view my order history and trade confirmations.	I can see a detailed history of my past orders and their execution status.	Medium	Sprint 2
Customer (Web user)	Portfolio Management	USM-8	As a user, I want to track my stock portfolio and its performance.	I can see a list of my owned stocks, their current value, and overall portfolio gain/loss.	High	Sprint 2
Customer (Web user)	Portfolio Management	USM-9	As a user, I want to set up watchlists to monitor specific stocks.	I can create custom watchlists and add/remove stocks as needed.	Medium	Sprint 3
Customer (Web user)	Research	USM-10	As a user, I want to access stock research and analysis.	I can view analyst ratings, company financials, and news articles related to stocks.	Medium	Sprint 3
Customer (Web user)	Customer Support	USM-11	As a user, I want to contact customer support for assistance.	I can submit a support ticket or access a knowledge base for troubleshooting.	Low	Sprint 4
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance Criteria	Priority	Release

3.4 Technology Stack

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

4. PROJECT DESIGN

4.1 Problem Solution Fit

1. CUSTOMER SEGMENTS <ul style="list-style-type: none"> • Active retail traders • Tech-savvy investors • Financial analysts using web platforms 	2. PROBLEMS <ul style="list-style-type: none"> • Need for real-time stock updates • Access platform on multiple devices • Ensure privacy 	3. TRIGGERS <ul style="list-style-type: none"> • Market volatility increasing real-time demand • Rising interest in online trading platforms • Awareness of data breaches
4. EMOTIONS: BEFORE / AFTER <ul style="list-style-type: none"> • Before: Anxious, distrustful, limited control • After: Confident, secure, empowered in real-time 	5. AVAILABLE SOLUTIONS <ul style="list-style-type: none"> • Traditional stock platforms • WebSocket APIs for updates • seamless UI and MERN integration 	6. CUSTOMER CONSTRAINTS <ul style="list-style-type: none"> • Low internet bandwidth during high trading hours • Budget limits for premium trading tools
7. BEHAVIOUR <ul style="list-style-type: none"> • Customers explore multiple apps • Customers explore multiple apps 	9. PROBLEM ROOT CAUSE <ul style="list-style-type: none"> • Lack of unified architecture for real-time sync • Insecure frameworks and poor compliance • Legacy Tech not designed for modern users 	10. YOUR SOLUTION <ul style="list-style-type: none"> • Build with MERN stack using Web Sockets • End-to-end encryption & role-based access • Scalable backend with microservices and clean UI/UX
8. CHANNELS OF BEHAVIOUR <ul style="list-style-type: none"> • Online: Trading blogs, YouTube reviews • Offline: Word-of-mouth, seminars, tech expos 		

4.2 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The application needs to handle real-time stock price updates, user transactions, and portfolio changes seamlessly while ensuring the security, privacy, and regulatory compliance of user data. The system currently lacks efficient real-time data handling, scalability, and robust security measures.
2.	Idea / Solution description	Develop a stock trading web application using the MERN stack, incorporating WebSockets for real-time data updates, scalable microservices architecture, and strong security protocols. The solution will include features like real-time stock price updates, user portfolio management, secure authentication, and compliance with financial regulations.
3.	Novelty / Uniqueness	The application will leverage the MERN stack's strengths to provide a seamless, real-time trading experience. Unique features include real-time synchronization using WebSockets, microservices for scalability, and advanced security measures like encryption and two-factor authentication.
4.	Social Impact / Customer Satisfaction	The solution aims to enhance the trading experience by providing timely and accurate information, thus enabling users to make informed decisions. Improved security and compliance will build user trust, leading to higher customer satisfaction and loyalty.
5.	Business Model (Revenue Model)	The application can generate revenue through subscription plans, transaction fees, premium features, and advertisements. Additional revenue streams can include partnerships with financial institutions and offering API access to third-party developers.
6.	Scalability of the Solution	The microservices architecture ensures that the application can scale horizontally by adding more instances as the user base grows. The use of cloud services for hosting and database management further supports scalability and flexibility to handle increased load and demand.

4.3 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture of Project:

1) Client:

- User: Represents the end-user who interacts with the stock trading web app.
- Stock Trading Web App: The front-end application accessed by the user through their browser.

2) Browser: The web browser used by the client to access the web application.

3) WebServer:

- Node.js: The JavaScript runtime environment used for building the server-side of the application.
- Express.js: A web application framework for Node.js, used to handle HTTP requests and route them to the appropriate handlers.

4) AppServer:

- React.js: The JavaScript library used for building the user interface of the web app. React components are rendered in the browser, providing a dynamic and responsive user experience.

5) Database:

- MongoDB: The NoSQL database used to store user information, stock data, transaction history, and other relevant data for the web app.

6) ExternalServices:

- Stock Market Data API: An external service used to fetch real-time stock market data, which is crucial for providing up-to-date information to the users.
- Notification Service: An external service used to send notifications to users about their transactions, stock price changes, or other important events.

Interaction Flow:

1) Client Interaction:

- The user interacts with the stock trading web app through their browser.

2) Request Flow:

- The browser sends an HTTP request to the WebServer (Node.js with Express.js) to access the web app.

3) User Actions:

- The user interacts with the React components in the app (e.g., checking stock prices, making transactions).
- These interactions result in API calls from the React app to the WebServer.

4) Data Operations:

- The WebServer handles these API calls and performs CRUD (Create, Read, Update, Delete) operations on the MongoDB database.

5) External Services Interaction:

- The WebServer fetches real-time stock data from the Stock Market Data API and sends notifications via the Notification Service when necessary.
- The fetched data is saved in the MongoDB database for future use and quick access.

6) Response Flow:

- The WebServer processes the data and returns the API responses to the React app.
- The React app updates the user interface based on the responses, providing the user with real-time data and feedback.

Notes:

- Client Note: The client interacts with the React.js application, which is served by the Node.js and Express.js backend.
- Database Note: MongoDB is used for storing user information, stock data, and transaction history.
- External Services Note: External services are used for fetching real-time stock market data and sending notifications to users.

Solution Architecture Diagram:

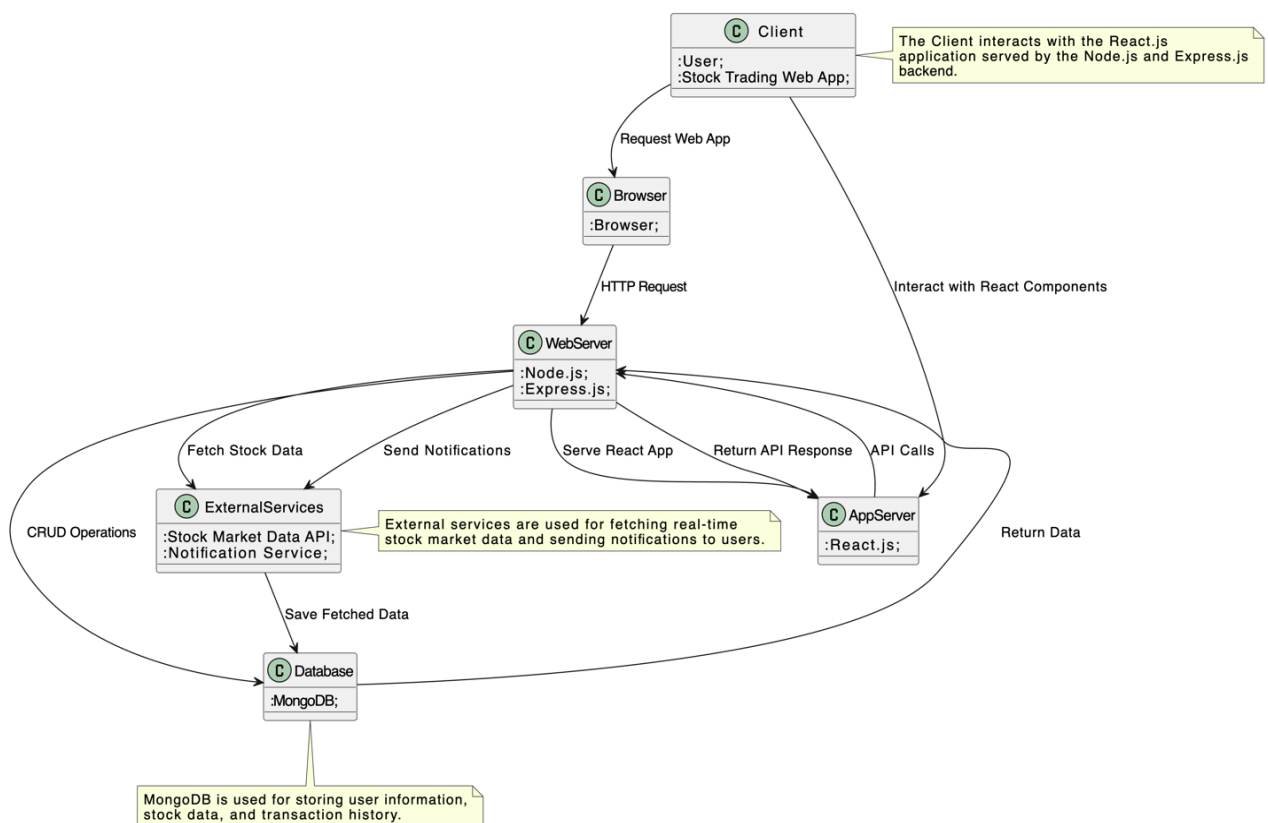


Figure 1: Architecture and data flow of the stock trading web application

5. PROJECT PLANNING & SCHEDULING

5.1 Project Planning

Product Backlog, Sprint Schedule, and Estimation (2 Marks)						
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User Registration, Confirmation	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	10	High	Nupur Trivedi
Sprint-1	User Login	USN-2	As a user, I can log into the application by entering email & password	10	High	Divyanshi
Sprint-2	User Portfolio Management	USN-3	As a user, I can view my portfolio and edit the portfolio	5	Medium	Mitul
Sprint-2	Stock Market Information	USN-4	As a user, I can view about the information of the stocks when needed	15	High	Vassu
Sprint-3	Trading Functionality	USN-5	As a user, I can sell buy or sell stocks	40	High	Mitul
Sprint-4	Alerts	USN-6	As a user, I can receive notification about the price changes and other news about the stocks	10	Medium	Divyanshi
Sprint-4	Reporting and Analytics	USN-7	As a user, I need to get information and statistic reports as the request is made. .	10	High	Nupur

6. FUNCTIONAL AND PERFORMANCE TESTING

6.1 Performance Testing

Overview:

A MERN stack-based online platform for real-time stock trading, offering user-friendly features for secure trading, portfolio management, and real-time stock data.

Key Features:

User Authentication: Secure registration, login, and password recovery.

Trading: Real-time stock prices, buy/sell orders, and trade history.

Portfolio Management: Track investments and receive notifications.

Technology Stack:

Frontend: React.js

Backend: Node.js, Express.js

Database: MongoDB

External Services: Stock data APIs, notification services

The app aims to deliver a seamless and secure trading experience with robust performance and scalability.

Project Version: Version 1

Testing Period: 8/04/25 to 10/04/25

Testing Scope:

List of Features and Functionalities to be Tested:

- User Registration and Login
- Viewing Real-Time Stock Data
- Placing Buy/Sell Orders
- Viewing Trade History
- Portfolio Management
- Receiving Notifications
- User Settings and Profile Management

List of User Stories or Requirements to be Tested:

- As a user, I want to register and log into my account.
- As a user, I want to view real-time stock prices.
- As a user, I want to place buy orders for stocks.
- As a user, I want to place sell orders for stocks.
- As a user, I want to view my trade history.
- As a user, I want to manage my stock portfolio.
- As a user, I want to receive notifications about my trades.
- As a user, I want to update my profile settings.

Testing Environment:

- **URL/Location:** http://localhost:3000/home
- **Credentials (if required):** [mukundan7610@gmail.com /

Test Cases:

<i>Test Case ID</i>	<i>Test Scenario</i>	<i>Test Steps</i>	<i>Expected Result</i>	<i>Actual Result</i>	<i>Pass/Fail</i>
TC-001	User Registration	1. Navigate to the registration page 2. Enter required details 3. Submit the form	User is successfully registered	User is successfully registered and redirected to the dashboard page	Pass
TC-002	User Login	1. Navigate to the login page 2. Enter username and password 3. Click on login	User is successfully logged in	User is logged in and redirected to the home page	Pass
TC-003	View Real-Time Stock Data	1. Log into the account 2. Navigate to the stock data page	Real-time stock data is displayed	Real-time stock data is displayed correctly	Pass
TC-004	Place Buy Order	1. Log into the account 2. Navigate to buy order page 3. Enter	Buy order is successfully placed	Buy order placed and confirmation message shown	Pass

		stock details 4. Submit the order			
TC-005	Place Sell Order	1. Log into the account 2. Navigate to sell order page 3. Enter stock details 4. Submit the order	Sell order is successfully placed	Sell order placed and confirmation message shown	Pass
TC-006	View Trade History	1. Log into the account 2. Navigate to trade history page	Trade history is displayed	Trade history is displayed correctly	Pass
TC-007	Manage Portfolio	1. Log into the account 2. Navigate to portfolio management page	Portfolio is displayed and manageable	Portfolio details are displayed and can be managed	Pass
TC-008	Receive Notifications	1. Perform trade actions 2. Check for notifications	Notifications are received	Notifications are received for trade actions	Pass
TC-009	Update Profile Settings	1. Log into the account 2. Navigate to profile settings page 3. Update settings 4. Save changes	Profile settings are updated	Profile settings updated successfully	Pass

Bug Tracking:

Bug ID	Bug Description	Steps to Reproduce	Severity	Status	Additional Feedback
BG-001	Error during user registration	1. Navigate to the registration page 2. Enter required details 3. Submit the form	High	Open	Registration fails with a server error
BG-002	Real-time stock data not updating	1. Log into the account 2. Navigate to the stock data page	Medium	In Progress	Stock data does not refresh automatically
BG-003	Unable to place buy order	1. Log into the account 2. Navigate to buy order page 3. Enter stock details 4. Submit the order	High	Closed	Buy order form submission results in an error
BG-004	Notifications not received	1. Perform trade actions 2. Check for notifications	Low	Open	Notifications not received after trade actions
BG-005	Profile settings not saving	1. Log into the account 2. Navigate to profile settings page 3. Update settings 4. Save changes	Medium	Open	Changes to profile settings are not saved

7. RESULTS

7.1 Output Screenshots

The screenshot displays the TradeSmart website. The top navigation bar includes the TradeSmart logo, links for Home, Markets, Products, Pricing, and Learn, and buttons for Log in and Register. The main headline reads "Smarter trading for modern investors". Below this, a sub-headline states: "Trade stocks, options, mutual funds, and more with zero brokerage fees. Access advanced charts, expert research, and powerful trading tools for both beginners and pros." Two buttons, "Get started" and "Learn more", are positioned below the text. A blue banner section titled "Trust with confidence" contains the text: "That's why 1.5+ crore customers trust TradeSmart with ₹4.5+ lakh crores of equity investments and contribute to 15% of daily retail exchange volumes in India." Below this banner, the "Customer-first always" section features four cards: "No spam or gimmicks", "The TradeSmart universe", "Do better with money", and "Smart Decision Making". The bottom section, "By the numbers", lists four key metrics: 15% Daily retail volume, ₹4.5+ Lakh Cr Equity investments, 1.5+ Cr Active customers, and 30+ Fintech startups.

TradeSmart Home Markets Products Pricing Learn Log in Register

Smarter trading for modern investors

Trade stocks, options, mutual funds, and more with zero brokerage fees. Access advanced charts, expert research, and powerful trading tools for both beginners and pros.

Get started Learn more

Trust with confidence

That's why 1.5+ crore customers trust TradeSmart with ₹4.5+ lakh crores of equity investments and contribute to 15% of daily retail exchange volumes in India.

TradeSmart Home Markets Products Pricing Learn Log in Register

Customer-first always

No spam or gimmicks

No gimmicks, spam, "gamification", or annoying push notifications. High quality apps that you use at your pace, the way you like.

The TradeSmart universe

Not just an app, but a whole ecosystem. Our investments in 30+ fintech startups offer you tailored services specific to your needs.

Do better with money

With initiatives like **Nudge** and **Kill Switch**, we don't just facilitate transactions, but actively help you do better with your money.

Smart Decision Making

Access detailed analytics and intelligent insights to help you make informed investment decisions based on market trends.

By the numbers

TradeSmart has become India's trusted trading platform

15%	₹4.5+ Lakh Cr	1.5+ Cr	30+
Daily retail volume	Equity investments	Active customers	Fintech startups

Welcome Back

×

Sign in to access your TradeSmart account

Email Address

✉ you@example.com

Password

🔒

👁

Forgot password?

☐ Keep me signed in for 30 days

Sign In

————— New to TradeSmart? —————

Create an Account

🔒 Secured by industry-leading encryption

Create Account

×

Join thousands of investors on TradeSmart

Account Type

Individual User

Admin Access

Username

👤 johndoe

Email Address

✉ you@example.com

PAN Card Number

📄 ABCDE1234F

Password

🔒

👁

Confirm Password

🔒

👁

☐ I agree to the [Terms of Service](#) and [Privacy Policy](#)

Create Account

WATCHLIST	SYMBOL ↑	NAME	PRICE	CHANGE	VOLUME	MARKET CAP	SECTOR	DETAIL
★	AAPL NASDAQ	Apple Inc.	\$212.06	↗ +3.51 (+2.00%)	78.95M	2.85T	Technology	View
★	AMZN NASDAQ	Amazon.com Inc.	\$209.87	↗ +2.35 (+1.30%)	32.46M	1.90T	Consumer Cyclical	View
★	GOOGL NASDAQ	Alphabet Inc.	\$196.69	↘ -1.98 (-1.37%)	18.98M	1.85T	Communication Services	View
★	HSBC LSE	HSBC Holdings plc	\$217.83	↗ +0.25 (+0.64%)	5.68M	78.00B	Financial Services	View
★	JPM NYSE	JPMorgan Chase & Co.	\$233.45	↗ +0.95 (+0.65%)	12.35M	427.00B	Financial Services	View
★	META NASDAQ	Meta Platforms, Inc.	\$220.36	↘ -0.75 (-0.23%)	15.69M	840.00B	Communication Services	View
★	MSFT NASDAQ	Microsoft Corporation	\$219.64	↗ +4.05 (+1.24%)	25.68M	2.45T	Technology	View
★	NVDA NASDAQ	NVIDIA Corporation	\$201.62	↗ +7.89 (+1.76%)	28.77M	1.12T	Technology	View
★	PG NYSE	The Procter & Gamble Company	\$233.05	↘ -0.32 (-0.20%)	6.54M	377.00B	Consumer Defensive	View

AAPL NASDAQ

\$212.15

-0.36 (-0.16%)

Volume

28.98M

Market Cap

\$715.20B

P/E Ratio

67.82

Sector

Automotive

1D

1W


1M

3M

1Y

5Y

16:18:41



Place Order

Real-time trading for AAPL

Buy

Sell

Current Price

\$212.15119

Product type

Intraday

Quantity

0

Total price

\$ 0.00

Buy now

< My Portfolio

Track your investments and performance

Overall profit

\$

\$49.22

Assets

1

Daily Change

+1.17%

Today

Q Search stocks...

Refresh

Stock	Price	Units	Total Value	24h Change	Actions
AAPL NASDAQ	\$209.00	20	\$4,195.08	↗ +1.17%	Trade

Transaction History

View all your deposits, withdrawals, buys and sells

Download

Filter

All

Deposits

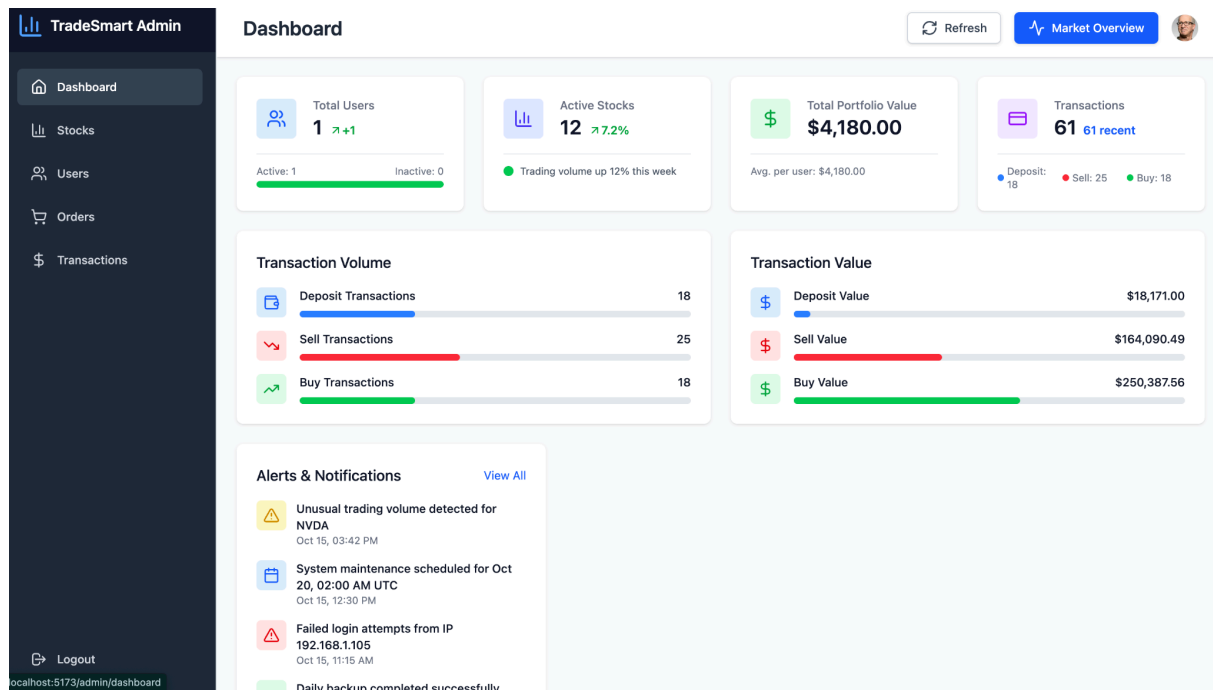
Withdrawals

Buy

Sell

Q Search transactions

TYPE	AMOUNT	PAYMENT MODE	STATUS	DATE
<div>🛒 Buy</div>	-\$4,195.08	<div>📁 stock</div>	completed	<div>📅 Apr 6, 2025, 04:19 PM</div>
<div>📈 Sell</div>	+\$2,316.94	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>📈 Sell</div>	+\$5,328.97	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>📈 Sell</div>	+\$22,753.26	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>📈 Sell</div>	+\$2,275.33	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>📈 Sell</div>	+\$4,550.65	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>📈 Sell</div>	+\$2,275.33	<div>📁 stock</div>	completed	<div>📅 Apr 5, 2025, 06:12 PM</div>
<div>🛒 Buy</div>	-\$20,844.91	<div>📁 stock</div>	completed	<div>📅 Apr 4, 2025, 09:33 PM</div>



8. ADVANTAGES & DISADVANTAGES

8.1 Advantages

- Easy and fast online stock trading
- Real-time stock price updates
- User-friendly interface for beginners and experts
- Portfolio tracking & performance insights
- 24/7 access from any device
- Secure transactions and user data protection
- Personalized watchlist and recommendations
- Saves time compared to traditional trading methods
- Instant transaction history and reports
- Accessible anytime, anywhere

8.2 Disadvantages

- There is a known issue with the real-time stock price updates occasionally lagging. We are working on optimizing the WebSocket connections to fix this.
- The application may experience slow performance under heavy load. Further optimization is planned.

9. CONCLUSION

The Stock Trading Web Application provides a smart, secure, and user-friendly platform for users to trade stocks online with ease. It simplifies the trading process by offering real-time market updates, portfolio management, and personalized insights all in one place. This application not only enhances the user experience but also makes stock trading more accessible, faster, and efficient for everyone — from beginners to expert traders.

10. FUTURE SCOPE

- Integration with additional stock exchanges.
- Mobile app development for iOS and Android.
- Enhanced data analytics and reporting features.
- Social trading features allowing users to follow and copy trades from successful traders.

11. APPENDIX

Documentation

- [MongoDB Documentation](#)
- [Express.js Documentation](#)
- [React Documentation](#)
- [Node.js Documentation](#)

Learning Resources

- [MERN Stack Tutorial](#)
- [JWT Authentication Tutorial](#)

- [React Router Documentation](#)
- [Redux Toolkit Documentation](#)

API Resources

- [Financial Modeling Prep API](#)
- [Alpha Vantage API](#)