Software Requirements Specification

for

PHInance

Version 1.00

Prepared by

Group 15 — **Group Name:** ***FEINance***

Aadi Singh	**220004**	**aadis22@iitk.ac.in**
Aadya Dhir	**220010**	**aadyad22@iitk.ac.in**
Anish Sahu	**220148**	**anishs22@iitk.ac.in**
Anisha Srivastava	**220149**	**anishas22@iitk.ac.in**
Dilbar Singh Lamba	**220368**	**dilbars22@iitk.ac.in**
Pranshu Thirani	**220801**	**pranshut22@iitk.ac.in**
Sameer Yadav	**220950**	**sameer22@iitk.ac.in**
Sangam Gupta	**220961**	**sangamg22@iitk.ac.in**
Aruz Awasthi	**230209**	**aruza23@iitk.ac.in**
Anirudh	**220146**	**anirudhm22@iitk.ac.in**

Course: **CS253**

Mentor TA: ***Namam Baranwal***

Date: **24/01/25**

Contents II

Revisions II

1 Introduction 1

1.1 Product Scope 1
1.2 Intended Audience and Document Overview 1
1.3 Definitions, Acronyms and Abbreviations 1
1.4 Document Conventions 1
1.5 References and Acknowledgments 2

2 Overall Description 2

2.1 Product Overview 2
2.2 Product Functionality 3
2.3 Design and Implementation Constraints 3
2.4 Assumptions and Dependencies 3

3 Specific Requirements 4

3.1 External Interface Requirements 4
3.2 Functional Requirements 4
3.3 Use Case Model 5

4 Other Non-functional Requirements 6

4.1 Performance Requirements 6
4.2 Safety and Security Requirements 6
4.3 Software Quality Attributes 6

5 Other Requirements 7

Appendix A – Data Dictionary 8

Appendix B - Group Log 9

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
v 1.00	Aadi Singh Aadya Dhir Anish Sahu Anisha Srivastava Anirudh Dilbar Singh Lamba Pranshu Thirani Sameer Yadav Sangam Gupta Aruz Awasthi	First version of the requirement document	24/01/26

1. Introduction

1.1. Product Scope

Managing financial investments and understanding stock markets can often seem daunting, particularly for beginners and individuals seeking a risk-free environment to experiment with trading strategies. Our project aims to bridge this gap by providing a simulated trading platform, catering to users who wish to explore stock trading in a safe and educational setting.

This website offers a simulated trading environment using virtual money for practice, empowering users to trade real-life stocks without the risk of monetary loss. In addition to live stock data, the platform provides comprehensive financial insights, technical analysis tools, and AI-powered predictions to support informed decision-making. Furthermore, features such as real-time portfolio management, mock IPOs, and behavioral analytics enhance the user experience and foster financial literacy. This website is not just a tool for learning but a gateway for users to experiment with trading strategies, assess risks, and grow their confidence in stock markets. The ultimate goal is to create a robust financial simulation ecosystem that supports users across all experience levels—providing advanced tools for seasoned traders and educational resources for beginners—while maintaining a user-friendly interface and secure system.

1.2. Intended Audience and Document Overview

1.2.1 Intended Audience:

- **Software Developers:** To utilize this document as a comprehensive guide for developing the website in alignment with the outlined requirements and functionalities.
- **Project Managers:** To oversee the development process, track milestones, and ensure the project adheres to its goals and timelines.
- **Testers and Approvers:** To validate the functionality and performance of the website against the defined requirements.
- **End Users:** Aspiring traders and individuals looking to explore and experiment with stock trading in a simulated environment.

1.2.2 Document Overview

1. Introduction: This section outlines the product scope and provides foundational information such as document conventions, abbreviations, and definitions. While experienced readers may opt to skip this section, it serves as a useful resource to resolve potential uncertainties encountered while reviewing the document.

2. Overall Description This section offers a high-level summary of the software, covering its primary objectives, functionality, and assumptions. It is recommended to read this section to gain an initial understanding of the system's purpose and context before diving into the details.

3. Specific Requirements This section presents a comprehensive breakdown of the software's features and requirements, illustrated with supporting diagrams. It is a critical

resource for developers during the implementation phase and provides end-users with a clear understanding of the system's capabilities.

4. Other Non-Functional Requirements: This section focuses on the non-functional requirements of the software, detailing performance, security, and usability standards. Developers will find this section particularly helpful for ensuring the system meets its operational and quality expectations.

5. Appendices: This section includes additional resources, such as appendices, to support readers with supplementary information, references, or data relevant to the project.

1.3. Definitions, Acronyms and Abbreviations

● **AI (Artificial Intelligence):** The broad field involving machines performing tasks that typically require human intelligence, such as decision-making and language processing.

● **AI Predictions:** End-of-day forecast generated by Machine Learning or Deep Learning models to predict next-day price movements.

● **API (Application Programming Interface):** A set of protocols and tools allowing different software components to communicate with each other.

● **Backend:** The server-side website or microservices that handle business logic, data processing, and communication with databases.

● **DB (Database):** A structured system for storing and retrieving data, such as user portfolios, transaction records, and market data.

● **Dividend Yield:** The ratio of a company’s annual dividend compared to its share price, expressed as a percentage.

● **DL (Deep Learning):** A subset of ML employing multi-layered neural networks to model complex patterns in data.

● **EOD (End of Day):** The time after the stock market closes; often used to trigger batch tasks such as model training or data aggregation.

● **Frontend:** The client-side interface (e.g., web or mobile app) with which the user interacts directly.

● **IPO (Initial Public Offering):** The process by which a private company offers shares to the public for the first time. (In this project, referred to as “Mock IPO” for educational simulation.)

● **Market Capitalization (Market Cap):** The total value of a company’s outstanding shares, calculated as share price multiplied by the number of shares.

● **ML (Machine Learning):** A subset of AI where algorithms learn patterns from historical data without being explicitly programmed for every outcome.

● **Mock IPO:** A simulated Initial Public Offering process within the platform, allowing users to learn how IPO subscriptions and allocations work.

● **P/E Ratio (Price-to-Earnings Ratio):** A popular valuation metric calculated by dividing a company's share price by its earnings per share (EPS).

● **PnL (Profit and Loss):** The gains or losses realized (or unrealized) from trading activities in the user's portfolio.

● **Portfolio:** A collection of stock (or other asset) holdings that a user manages within the platform.

● **Real-Time Data:** Live or near-live stock market updates (price and volume) the platform uses to simulate actual trading conditions.

● **Risk/Reward Ratio:** A measure comparing the potential profit of a trade to its possible loss, giving users insight into trade viability.

● **ROI (Return on Investment):** A performance measure used to evaluate the efficiency or profitability of an investment, usually expressed as a percentage.

● **Screener:** A tool that filters stocks based on user-defined criteria (e.g., P/E ratio, dividend yield, market cap).

● **Trade:** A buy or sell transaction initiated by the user in the simulated environment.

● **Value at Risk (VaR):** A risk management technique estimating the potential loss in a portfolio over a defined time frame with a given confidence level.

● **Virtual Funds (Fake Money):** The simulated currency allocated to each user's account, used to practice trading without risking real capital.

1.4. Document Conventions

- Arial font size 11 is used throughout the document for text.
- Arial font sizes of 14 and 18 are used for Subheadings and headings, respectively.
- The headings and subheadings of all sections are written in bold.
- Bullet point ordering has been used as a listing typesetting tool.

1.5. References and Acknowledgments

- *https://notegpt.io/ai-flowchart-generator*
- *https://www.canva.com/*
- *https://hackmd.io/?nav=overview*

2. Overall Description

2.1. Product Overview

Catering to the recent boom in stock trading activity and interest among people, PHInance is designed to provide users with a simulated trading platform where they can experience the stock market in a risk-free environment and hone their trading instincts. The website replicates real-world trading scenarios using real-time stock data, allowing users to practice and refine their trading strategies. It serves as an educational tool for beginners while offering advanced features for experienced users to experiment with various financial instruments and analytics.

Whether users are complete novices or seasoned traders, this platform bridges the gap between theoretical knowledge and practical experience by offering a hands-on approach to understanding the intricacies of trading and portfolio management. The website is an all-in-one solution that educates, empowers, and enables users to navigate the stock market with confidence.

2.2. Product Functionality

The major functions of the website include:

- **Simulated Trading Environment:** Allowing users to perform simulated trades in a risk-free environment.

- **Comprehensive Stock Data:** Displaying comprehensive stock data, including financial metrics, sentiment analysis, market performance, market capitalization and other statistics about stocks.

- **Technical Analysis Tools:** To help users understand market movements and make informed decisions, we will provide technical analysis features for each stock, including On-Balance Volume (OBV), Accumulation/Distribution (A/D) Line, Average Directional Index (ADX), Aroon Oscillator, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Stochastic Oscillator, etc.

- **Generating AI-powered Predictions** for next-day price movements based on historical and real-time data.

- **Assisting Beginners:** Enabling users to manage virtual portfolios with insights into performance metrics like ROI and diversification.

- **Risk Analysis Tools:** Incorporating risk analysis tools, such as Value at Risk (VaR) and trade-specific risk/reward ratios.

- **Financial Education:** Promoting financial literacy through educational tools like videos, quizzes, and challenges.

● **Real-Time Data Integration:** Providing real-time or near-real-time data updates to replicate a live trading experience.

● **Behavioural Analytics:** Track user behavior (e.g., impulsive trading or frequent high-risk decisions) and provide personalized feedback to improve their trading habits.

2.3. Design and Implementation Constraints

- **Technological Stack:** The platform is built using PostgreSQL, Golang (with Gin/Gorm), and Next.js, limiting the use of alternative tools or frameworks.
- **Real-Time Data:** The system relies on real-time stock data integration, requiring low latency and stable external APIs.
- **Scalability:** The platform must support multiple concurrent users without performance degradation.
- **Network Dependency:** A stable internet connection is essential for real-time updates and trading simulations.
- **API Requirements**: The system must integrate with third-party APIs for real-time or near-real-time stock data.
- **ML Models:** Machine learning models for stock predictions must be trained on historical and real-time data. It is important to ensure models are lightweight and efficient to avoid delays in generating predictions.

2.4. Assumptions and Dependencies

- **Training Data:** Historical and real-time data will be sufficient to train accurate AI/ML models for stock predictions
- **Stable Internet Connection:** Users are expected to have a reliable internet connection for real-time stock data and seamless platform functionality.
- **Third-Party API Reliability:** The platform depends on external APIs for stock data, sentiment analysis, and news updates. Any disruptions in these services could impact functionality.
- **Modern User Devices:** The platform assumes users will access it via modern devices with supported web browsers for optimal performance.

3. Specific Requirements

3.1. External Interface Requirements

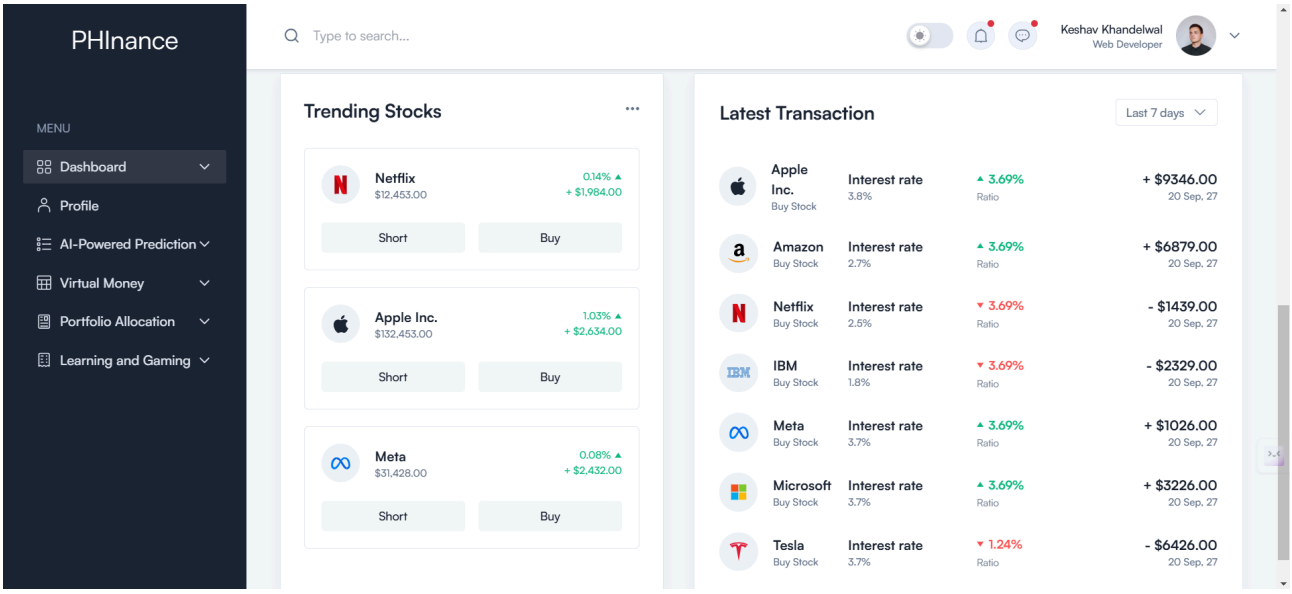
3.1.1. User Interfaces

1. Sign In/Sign Up Page

The login window prompts users to enter their email or phone number along with their password. After inputting these details into the designated text fields, users can click the "Sign In" option at the bottom to access their account. In case a user forgets their password, they can utilize the "Forgot Password?" option for recovery via OTP verification.

For new users, selecting the "Sign Up" option opens a window where they can establish a new account and set a password. The creation of a new account necessitates the completion of various credentials, including the owner's name, store/firm name, email, and phone number. Following this, OTP verification for both email and phone number is conducted. Subsequently, users can create a password adhering to predefined rules and confirm it.

2.Dashboard



Main Dashboard Sections

The main dashboard will be divided into multiple sections to provide a quick overview of trading performance, market updates, and available actions.

A. Portfolio Summary (Top Section)

Displays a snapshot of the user's portfolio with:
Virtual Balance: ₹10,000 (or updated balance)
Total Portfolio Value
Total ROI (Return on Investment)
Daily PnL (Profit/Loss)
Risk Level Indicator (e.g., Low, Medium, High, based on trade history & volatility)
Quick Action Buttons:
🛒 Buy Stocks
📉 Sell Stocks
📜 View Transaction History

B. Market Overview (Stock Performance Section)

Stock Market Indices: NIFTY, SENSEX, NASDAQ, S&P 500
Trending Stocks: Top gainers, top losers, and most active stocks
Stock Heatmap: A graphical representation of stock movements across various sectors
Sentiment Analysis Widget: Shows whether market sentiment is bullish, neutral, or bearish based on aggregated news and AI analysis

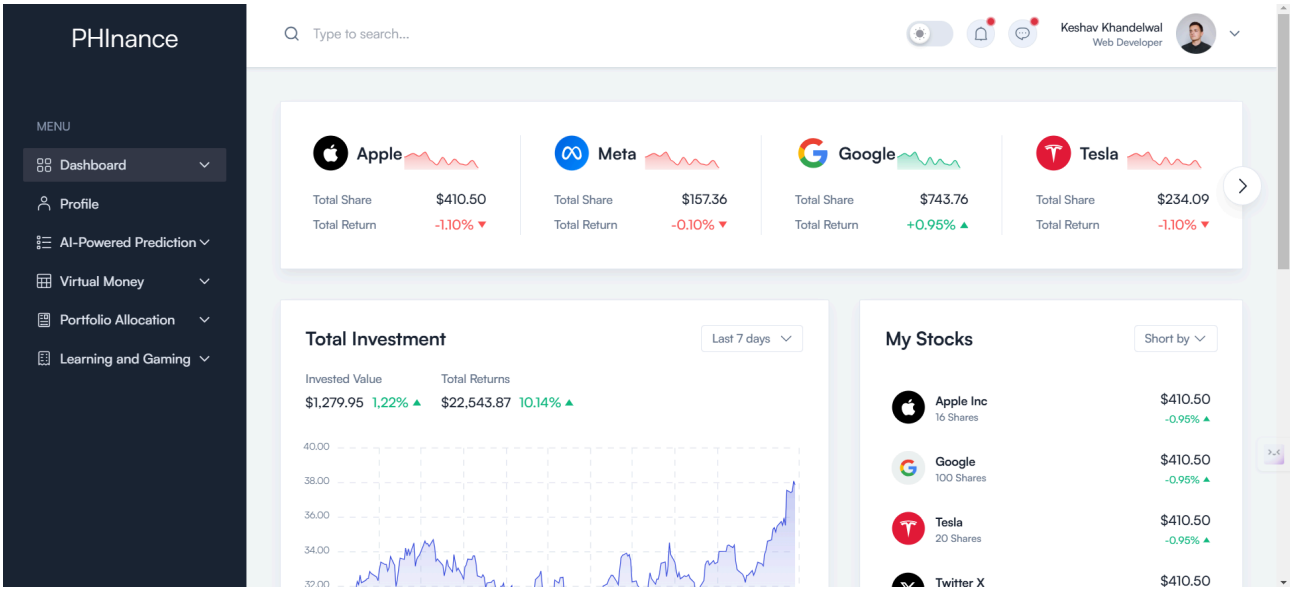
C. AI-Powered Predictions (End-of-Day Insights)

Tomorrow's Predicted Stock Trends with:
Stock Name → 🔼 or 🔽 (Up/Down Prediction)
Confidence Score (%)

Top Factors Influencing Prediction (e.g., RSI, MACD, Sentiment, Earnings Reports)
Users can click 'See Detailed Analysis' to explore model explanations and historical accuracy of AI predictions.

D. Watchlist & Favorite Stocks

Users can create and track their customized watchlist
Each stock card will display:
Current price & daily change (%)
52-week high/low
Technical indicator status (e.g., RSI Overbought, MACD Bullish Crossover)



E. Technical Analysis & Stock Screener

Interactive stock screener: Users can filter stocks based on:
P/E Ratio
Market Cap
Dividend Yield
Price Change (1D/1W/1M)
Technical Indicator Dashboard: Quick signals for RSI, MACD, ADX, Aroon Oscillator

F. Behavioral Analytics & Risk Management

Trade Behavior Summary:
Impulse trading alerts 📢
Trade consistency evaluation (e.g., high-frequency trades, panic selling detected)
Portfolio sector exposure insights
Value at Risk (VaR) Estimation:
Displays maximum expected loss for the next day under normal market conditions
Risk-Adjusted Recommendations:
Suggests if the portfolio is too risky and provides learning materials to improve strategies

G. Learning & Gamification Section

"Improve Your Skills" Module:

Watch Educational Videos (E.g., “Understanding MACD”)
Take a Trading Quiz (Earn extra virtual money)
Join Market Challenges
Completion Rewards: Users earn bonus virtual funds to continue practicing trading.

3. Footer (Global Information & Support)
About Us | FAQs | Contact Support | Terms & Conditions | Privacy Policy

3.1.2. Hardware Interfaces

- A device with a linux-based operating system environment, to host the server-side components, including the database.

3.1.3. Software Interfaces

- The server-side components, including the database, must be hosted in a Linux-based operating system environment.

- The client-side components must be functional on modern web browsers as well as their mobile versions, like Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, and Brave.

- The database management system used by us will be PostgreSQL. The following databases will be created:
    - The user database will store all users’ data, including authentication details and personal information - id, name, email, phone no, password, dob, balance, verification status, the date and times at which the user was created and/or updated.

    - The Ticker database will store the id of the company, name of the company and stock ticker symbol.

    - The Transaction database will store details about all the transactions made. It includes the transaction id, user id of the user who bought the stock, stock id, date and time of the transaction, volume, stock price and the transaction type (if the user is buying or selling).

    - The OTP database will store the user id, the OTP, OTP creation time and OTP expiry time.

3.2. Functional Requirements

3.2.1. F1: Sign in/ Sign up:

- The user should be able to sign in using their **email ID (or phone number) and password** if they are already registered.
- There should be an option for **'Forgot Password?'** in case the user forgets their credentials. Authentication shall be carried out through **OTP verification** on the registered email or phone number.
- New users should be able to register by providing their name, email, phone number, and a secure password. An OTP-based verification will be required to activate the account.

3.2.2. F2: Dashboard:

- The **dashboard** of the simulated trading platform will serve as the **central hub** where users can track their portfolio,buy and sell stocks , monitor stock movements, access AI-powered predictions, and explore learning resources. The design will be **user-friendly, intuitive, and data-driven**, incorporating real-time stock data, trading insights, and risk management tools.
    -
- **1. Header / Navigation Bar (Persistent at the Top)**
- **Platform Logo** (Clickable, redirects to Dashboard)
- **Search Bar** (Find stocks quickly by entering stock symbols or company names)
- **Navigation Links**:
- 📊 **Dashboard** (Home)
- 📂 **Portfolio**
- 🔍 **Risk Analysis**
- 🤖 **AI Predictions**
- 🎓 **Quizzes & Learning**
- ⚙️ **Profile / Settings**

Main Dashboard Sections

The main dashboard will be divided into multiple sections to provide a **quick overview of trading performance, market updates, and available actions.**

3.2.3. F3: Portfolio Summary :

- The Portfolio Summary section provides an at-a-glance overview of the user's financial standing within the simulated trading environment. This section enables users to quickly assess their performance, risk exposure, and available actions.

Key Metrics Displayed

- **Virtual Balance: ₹10,000 (or updated balance after trades)**
  - Represents the amount of virtual currency available for trading.
  - Updates dynamically as users execute buy/sell orders.
- **Total Portfolio Value:**
  - The current market value of all holdings, including cash and investments.
  - Formula:Portfolio Value=Cash Balance+∑(Shares Held×Current Market Price)Portfolio Value=Cash Balance+∑(Shares Held×Current Market Price)
- **Total ROI (Return on Investment):**
  - Measures the user's overall profitability.
  - Formula:ROI(%)=(Total Portfolio Value−Initial BalanceInitial Balance)×100ROI(%)=(Initial BalanceTotal Portfolio Value−Initial Balance)×100
  - A positive ROI means profitable investments, while a negative ROI indicates losses.
- **Daily PnL (Profit and Loss):**
  - Tracks the user's gains/losses for the current trading day.
  - Formula:Daily PnL=∑(Shares Held×Price Change for the Day)Daily PnL=∑(Shares Held×Price Change for the Day)
  - Helps users assess daily market movements and trading success**.**

Quick Action Buttons

These buttons allow users to interact with their portfolio efficiently:

- **Buy Stocks**
  - Opens a trading window to purchase stocks using available virtual funds.
  - Allows input of stock symbol, quantity, and order type (Market/Limit).
- **Sell Stocks**
  - Enables selling of currently held stocks.
  - Displays real-time market price and potential PnL before execution.
- **View Transaction History**
  - **Shows a detailed ledger of past trades including:**
    - Stock name & symbol
    - Date & time of transaction
    - Buy/Sell price per share
    - Total trade value
    - PnL from each trade

3.2.4. F4: Market Overview (Stock Performance Section):

- The **Market Overview** section provides **real-time insights** into stock performance, helping users make informed trading decisions. It includes **key stock market indices, trending**

stocks, sector movements, and sentiment analysis, giving users a **holistic view** of the market.

1.Stock Market Indices

This section tracks major stock indices from global and domestic markets, providing users with a **benchmark** to compare overall market trends.

- **NIFTY 50 (India)** – Represents the top 50 companies listed on the **National Stock Exchange (NSE)**.
- **SENSEX (India)** – Tracks the performance of **30 major stocks** on the **Bombay Stock Exchange (BSE)**.
- **NASDAQ (USA)** – A technology-focused index representing major tech and growth companies.
- **S&P 500 (USA)** – A widely-followed index representing the **500 largest U.S. companies**.

Display Features:

- Real-time percentage change (📈 Up or 📉 Down).
- Daily price movement and trading volume.
- Historical chart for trend analysis.

2. Trending Stocks

Displays the most active and volatile stocks, categorized into different lists:

- **Top Gainers:**
  - Stocks with the **highest percentage increase** in price today.
  - Indicates strong **bullish momentum** and positive market sentiment.
  - Useful for **momentum traders** looking for short-term opportunities.
- **Top Losers:**
  - Stocks experiencing the **highest percentage drop** in price today.
  - Helps traders identify **oversold stocks** or avoid stocks in **downtrends**.
- **Most Active Stocks:**
  - Stocks with the **highest trading volume** today.
  - Often represents stocks with **news-driven movement** or **high liquidity**.

3.2.5. F5: AI-Powered Predictions (End-of-Day Insights)

- The AI-Powered Predictions section provides users with next-day stock movement forecasts based on machine learning (ML) and deep learning (DL) models. This section helps traders anticipate market trends using historical price data, technical indicators, sentiment analysis, and financial metrics.

1. Tomorrow's Predicted Stock Trends

At the end of each trading day, the AI model generates predictions for the next trading session. These insights include:

- Stock Name & Symbol: Displays the stock ticker (e.g., TCS, INFY, AAPL, TSLA).

- Predicted Direction (🔼 Up / 🔽 Down): Indicates whether the stock is expected to rise or fall the next day.
- Confidence Score (%): Shows the AI model's probability estimation of its prediction being correct.
    - Example:
        - TCS 🔼 (Confidence: 72%) – High likelihood of price increase.
        - INFY 🔽 (Confidence: 65%) – Stock may decline based on trends.

Display Features:

- Sortable Table: Users can sort predictions based on confidence level, stock symbol, or expected price movement.
- Filtering Options: View only bullish/bearish stocks or filter by confidence score threshold (e.g., predictions with ≥ 70% confidence).

3.2.6. F6:Learning & Gamification Section :

- The **Learning & Gamification Section** is designed to **educate users, improve their trading skills, and incentivize engagement** through quizzes, videos, and market challenges. By participating in these activities, users can **earn extra virtual money** to continue trading in the simulated environment.

1. "Improve Your Skills" Module

This module provides **interactive learning resources** to help users **develop a strong foundation** in stock trading, technical analysis, and risk management.

Watch Educational Videos

- **Concept-based tutorials covering:**
    - **Technical Indicators** (e.g., "Understanding MACD," "How RSI Works")
    - **Fundamental Analysis** (e.g., "Interpreting P/E Ratio," "How to Read an Income Statement")
    - **Trading Strategies** (e.g., "Momentum vs. Value Investing," "Risk Management Techniques")
    - **Market Psychology** (e.g., "Avoiding Emotional Trading," "Behavioral Biases in Investing")

Features:

- Videos are **short and engaging (5-10 minutes).**
- Users **earn virtual funds (₹500-₹1,000)** after completing a video.
- A **progress tracker** monitors completed content.

Take a Trading Quiz

- Quizzes help users **test their knowledge** on key trading concepts.
- Topics include:
  - **Basic Trading Concepts** (e.g., "What is a stock?")
  - **Technical & Fundamental Analysis** (e.g., "What does a bullish MACD crossover indicate?")
  - **Risk & Money Management** (e.g., "How does Value at Risk (VaR) work?")

Features:

- Users can **take quizzes after watching related videos**.
- Multiple-choice format with **instant feedback and explanations**.
- **Passing a quiz rewards virtual funds (₹1,000-₹2,000)**.
- **Leaderboard** ranks users based on quiz performance.

2. Completion Rewards & Progress Tracking

- Users **earn additional virtual funds** for:
  - Watching a full video (+₹500)
  - Passing a quiz (+₹1,000)
  - Completing a trading challenge (+₹2,000-₹5,000)
- A **progress tracker** monitors completed videos, quizzes, and challenges.

3.2.7. F7: Frequently Asked Questions

- This feature of the software shall provide answers to the user's various questions and doubts that may arise while using the software.
- It shall serve as a quick reference guide to the user for using the software.

3.2.8. F8 : Technical Analysis:

The platform provides **real-time** and **historical technical indicators** to help users analyze stock trends.

Available Indicators

- **Moving Averages (SMA, EMA)**
  - Identifies short-term and long-term trends.
  - Example: **50-day SMA crossing above 200-day SMA → Bullish signal**.
- **MACD (Moving Average Convergence Divergence)**
  - Used to detect **momentum shifts**.

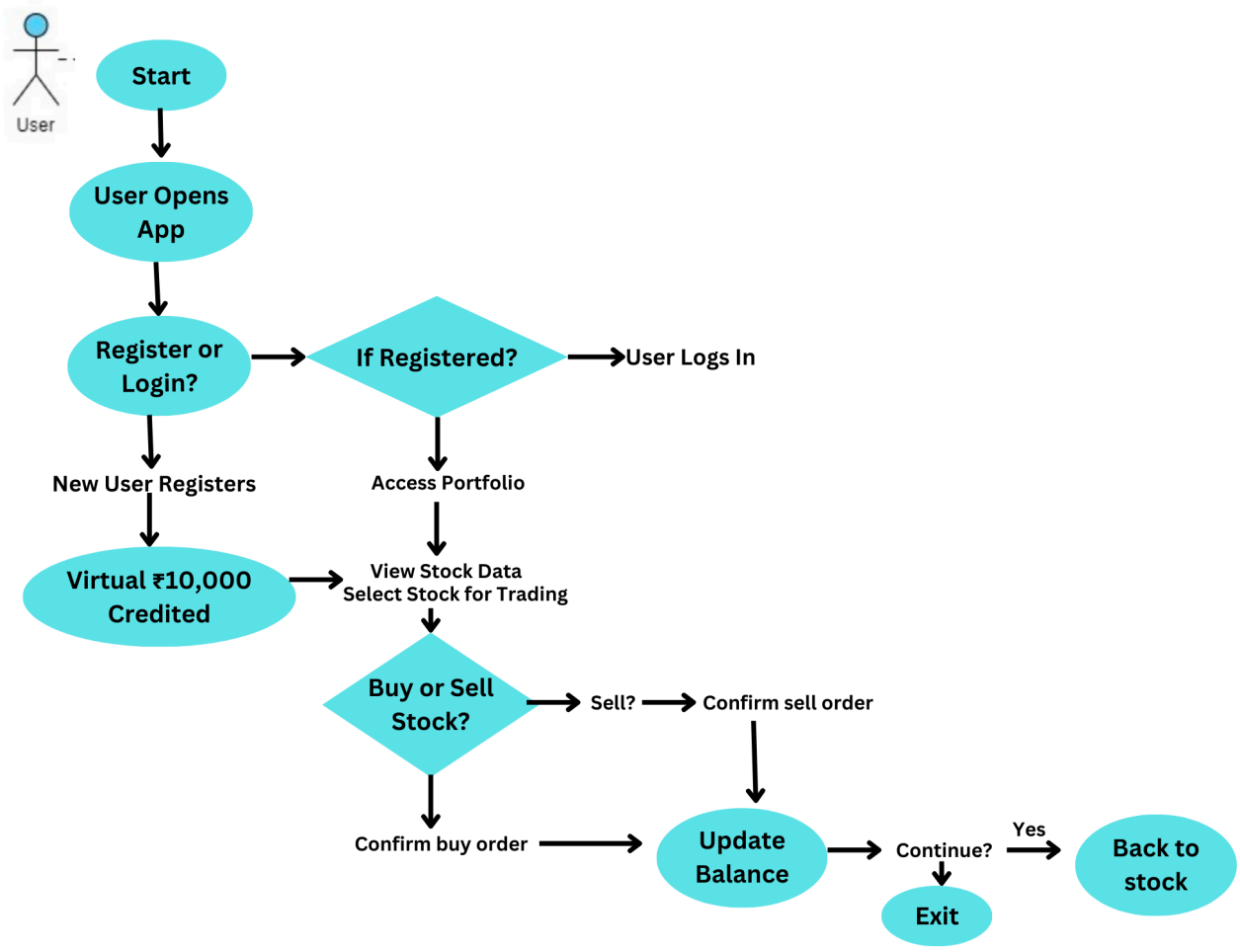
    - Example: **MACD line crossing above signal line → Bullish trend**.
- **RSI (Relative Strength Index)**
    - Measures **overbought (>70) or oversold (<30) conditions**.
    - Example: **RSI above 70 → Possible reversal or overvaluation**.
- **Bollinger Bands**
    - Shows **volatility levels** (Expanding bands = High volatility).
    - Example: **Stock touching upper band → Potential overbought signal**.
- **ADX (Average Directional Index)**
    - Measures **trend strength** (Above 25 = Strong trend, Below 20 = Weak trend).

📊 **Display Features:**

- **Interactive Charts** – Users can overlay indicators **on stock charts**.
- **Real-Time Updates** – Indicators update with **live market movements**.
- **Backtesting Feature** – Users can **apply past data to test strategies**.

3.3. Use Case Models

3.3.1. Use Case #1 Simulated Trading Environment:



Authors: Aadi Singh and Aadya Dhir

1 Purpose: Let users practice trading with virtual money. They receive ₹10,000 to buy or sell stocks, observe profits/losses, and learn in a risk-free setting.

2. Requirements Traceability

- **REQ-1**: New users start with ₹10,000 virtual balance.
- **REQ-2**: Stock data is updated in real time (or near real time).
- **REQ-3**: Buy/sell orders adjust user balances and portfolios.
- **REQ-4**: Users can't buy if they lack enough funds or sell if they lack enough shares.
- **REQ-5**: Transactions, balances, and portfolios update immediately.

3. Priority

High – Core feature of the platform and critical to user experience.

4. Preconditions

- User is **logged in** or **registered**.
- User has a **virtual balance** assigned.
- Stock market data is **available**.
- Enough funds for buy orders; enough shares for sell orders.

5. Post Conditions

- **Balance** updated after each trade.
- **Portfolio** reflects the new holdings.
- **Transaction record** created (price, time, etc.).
- If user runs out of money, platform prompts them to earn more via quizzes or videos.

6. Actors

- **User**: Initiates buy/sell orders.
- **System**: Processes trades and updates data.

7. Exceptions

1. **Insufficient Balance**: Can't buy.
2. **Insufficient Shares**: Can't sell.
3. **Data Retrieval Failure**: Stock data not available.
4. **Session Timeout**: User logged out before finishing.
5. **Trade Limits**: If daily or max trade limits exist, trades may be blocked.

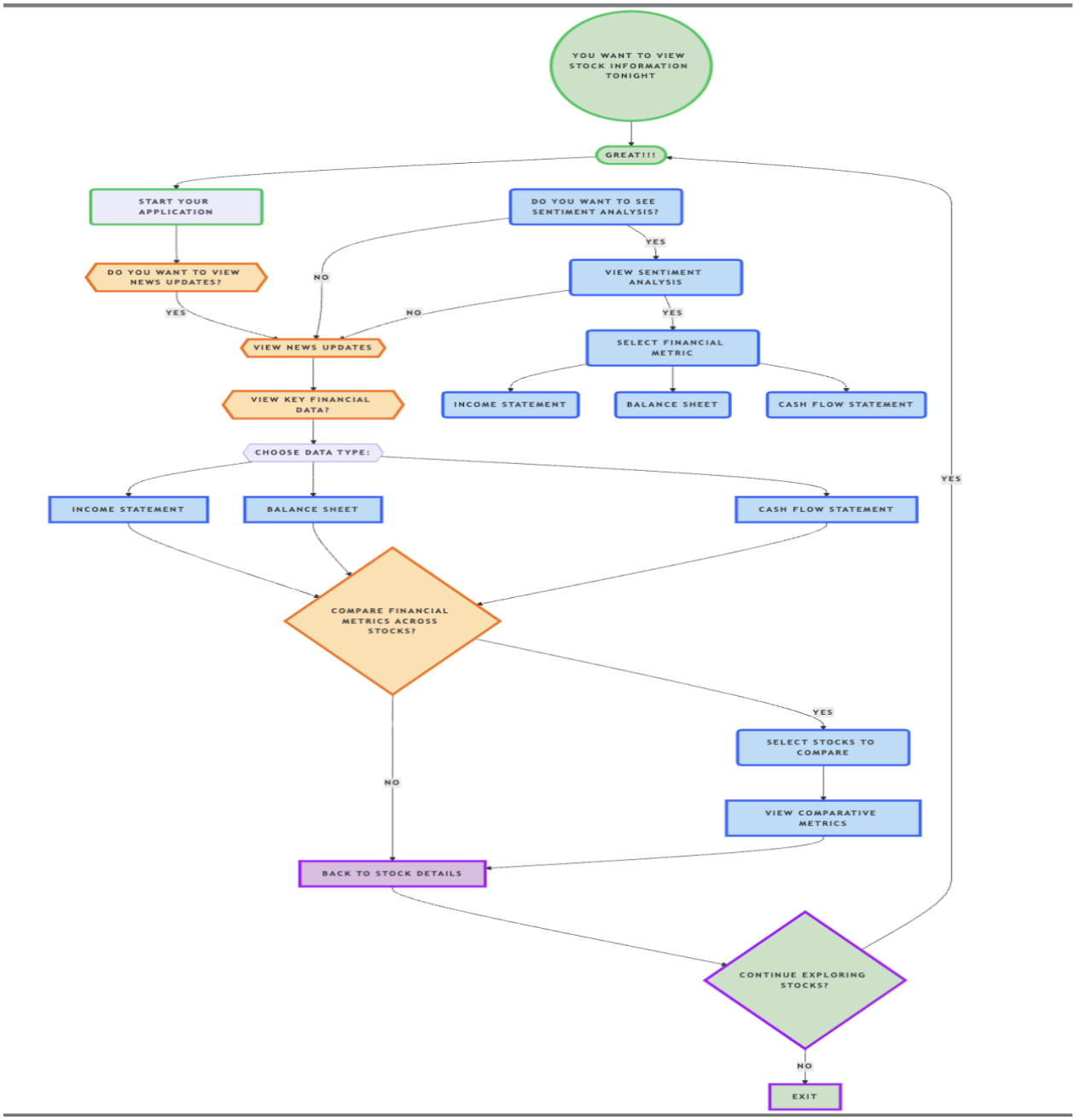
8. Includes

- **User Authentication** (login/registration).
- **Portfolio Management** (show updated holdings).
- **Real-Time Data Fetch** (stock prices).

9. Notes/Issues

- Decide if **partial shares** are allowed.
- Clarify if **limit orders** or only current market price.
- Handle data lag or downtime gracefully.

3.3.2. Use Case #2 Comprehensive Stock Data:



Use Case: View Comprehensive Stock Details

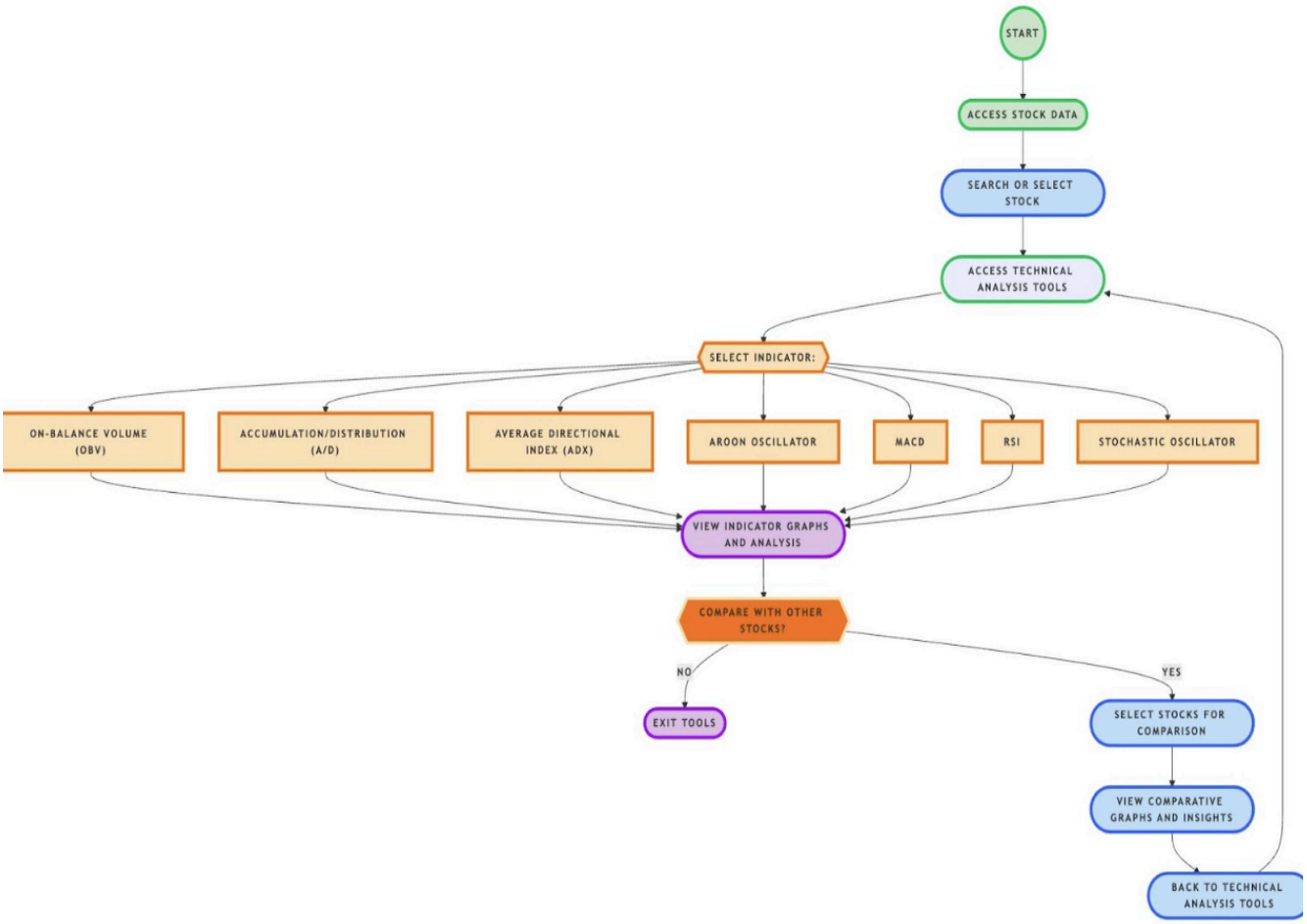
Authors: Dilbar and Anish

1. **Purpose**
   Let users see a **stock's complete information** (news, sentiment, financial statements) and compare key metrics across multiple stocks if needed.

2. **Requirements Traceability**
   - Show **detailed stock data** (e.g., price, news, sentiment).
   - Display **financial statements** (Income, Balance Sheet, Cash Flow).
   - Allow **comparison** across different stocks.
   - Provide **navigation** back to the portfolio or main list.
3. **Priority**
   High – Crucial for informed decision-making in the simulated environment.

4. **Preconditions**
   - User is **logged in**.
   - A **stock** is selected to view.
   - The system can **access** current data (news/financials).
5. **Post Conditions**
   - User **views** complete stock details.
   - User can **compare** metrics across multiple stocks.
   - User can **return** to other sections or exit.
6. **Actors**
   - **User**: Requests details.
   - **System**: Fetches and shows data.
7. **Exceptions**
   - **Data Not Available**: Missing news or financial info.
   - **API Failure**: Can't load real-time data.
   - **Session Timeout**: User is logged out mid-view.
   - **Comparison Limit**: Too many stocks to compare at once.
8. **Includes**
   - **User Authentication**
   - **Portfolio Management**
   - **Market Data Retrieval**
   -
9. **Notes/Issues**
   - Decide **refresh frequency** for news/sentiment.
   - Confirm max **compare limit** for stocks.
   - Ensure **UI** is user-friendly and responsive.

3.3.3. Use Case #3 Technical Analysis Tools:



Use Case: Technical Analysis Tools

Authors: Sameer and Anisha

1. **Purpose**
   Let users **view** and **compare** technical indicators (e.g., MACD, RSI, ADX) on selected stocks to analyze market trends.

2. **Requirements Traceability**

   - Provide **indicator options** (OBV, A/D, ADX, Aroon, MACD, RSI, Stochastic).
   - Show **graphs** of each selected indicator.
   - Allow **comparison** across stocks if desired.
   - Return to stock details or exit the tools.

3. **Priority**
   High – Essential for market analysis and user education.

4. **Preconditions**

   - User is **logged in**.
   - A **stock** is chosen for analysis.
   - Historical data is **available** for computing indicators.

5. **Post Conditions**

   - User has **visualized** one or more indicators.
   - User can optionally **compare** them across multiple stocks.
   - User can **return** to other sections or exit.

6. **Actors**

   - **User**: Initiates indicator selection/comparison.
   - **System**: Retrieves data, computes indicators, displays charts.

7. **Exceptions**

   - **Insufficient Data**: Not enough history for some indicators.
   - **API Failure**: Data feeds unavailable.
   - **Comparison Limit**: Too many stocks requested.
   - **Session Timeout**: User logs out mid-analysis.

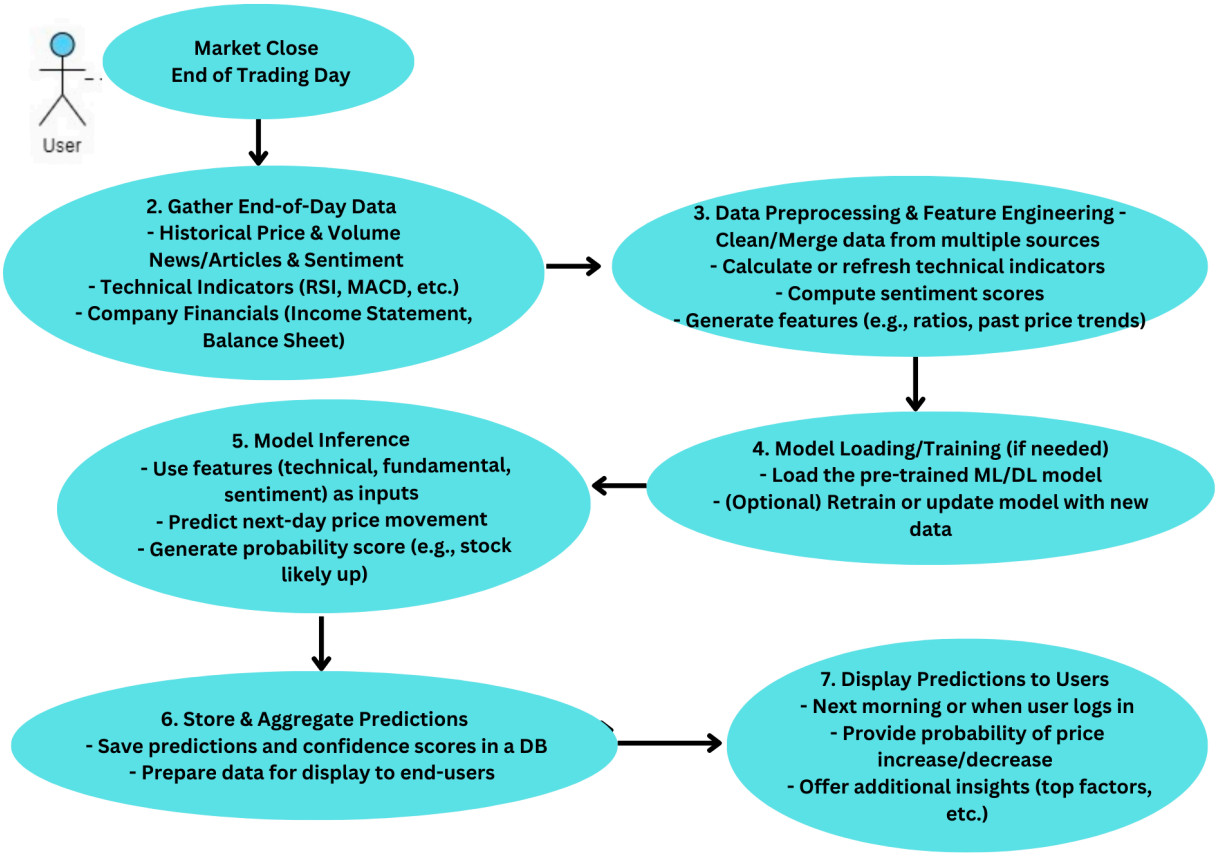
8. **Includes**

   - **Market Data Retrieval** (historical prices, volumes).
   - **User Authentication** (login).
   - **Stock Details** (navigation).

9. **Notes/Issues**

   - Decide **update frequency** for real-time vs. on-demand.
   - Ensure **UI** supports multiple charts side by side.

3.3.4. Use Case #4 AI-Powered Predictions:



Use Case: AI-Powered Predictions

Authors: Anirudh and Sangam

1. **Purpose**
   Generate **next-day stock price predictions** using ML/DL models, offering probability scores and insights to guide users' trading decisions.

2. **Requirements Traceability**

   - Collect **end-of-day data** (prices, volume, sentiment, technicals, financials).
   - **Preprocess** and merge data from multiple sources.
   - **Load** or retrain the ML/DL model with fresh data.
   - Produce **next-day movement predictions** and probability scores.
   - **Store** predictions in a database.
   - Display **results** to users, including confidence scores and key insights.

3. **Priority**
   High – Core feature providing significant value to users.

4. **Preconditions**

   - System can **access** end-of-day market data.
   - A **trained model** is available or can be retrained.
   - Sufficient **historical data** exists for feature engineering.

5. **Post Conditions**

   - **Predictions** (with probabilities) are generated and saved.
   - Users can **view** these forecasts with relevant confidence levels and insights.

6. **Actors**

   - **System** (AI/ML Engine): Gathers data, runs the model, saves outcomes.
   - **User**: Consumes the prediction results.

7. **Exceptions**

   - **Data Gaps**: Missing or corrupted data.
   - **Model Errors**: Failure to load or retrain.
   - **Storage Failures**: Unable to save predictions.
   - **Timing Delays**: Predictions run late if data arrives late or the model is slow.

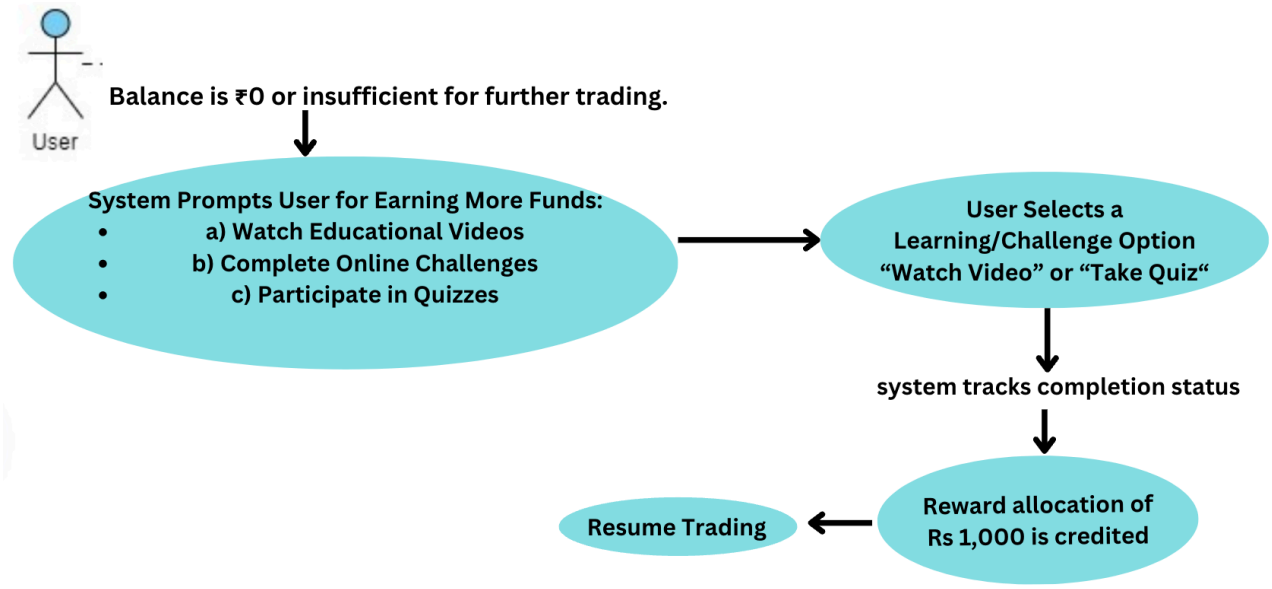
8. **Includes**

   - **Market Data Retrieval**
   - **Comprehensive Stock Details**
   - **User Authentication**

9. **Notes/Issues**

   - Determine **frequency** of retraining (daily, weekly, rolling).
   - Track **model performance** over time.
   - Provide **explainability** (top factors or indicators) for user trust.

3.3.5. Use Case #5 Educating Beginners



Use Case: Educating Beginners (Earning More Virtual Funds)

Authors: Aruz and Pranshu

1. **Purpose**
   Help users with low or zero balance regain trading funds by completing educational tasks (videos, quizzes, challenges).

2. **Requirements Traceability**

   - Detect when balance is too low.
   - Offer educational content (videos, quizzes).
   - Track completion.
   - Reward successful completion with additional funds.
   - Allow repeated participation if balance depletes again.

3. **Priority**
   High – Essential to keep users learning and trading.

4. **Preconditions**

   - User's balance is depleted.
   - Educational content is accessible.
   - User is logged in.

5. **Post Conditions**

   - User completes a learning activity.
   - System adds virtual funds.

- User can resume trading or continue learning.

6. **Actors**

    - **User**: Starts the educational activity.
    - **System**: Prompts, tracks, and rewards.

7. **Exceptions**

    - **Incomplete Activity**: No reward given.
    - **Network Failure**: Content not loaded.
    - **Session Timeout**: User logged out mid-activity.

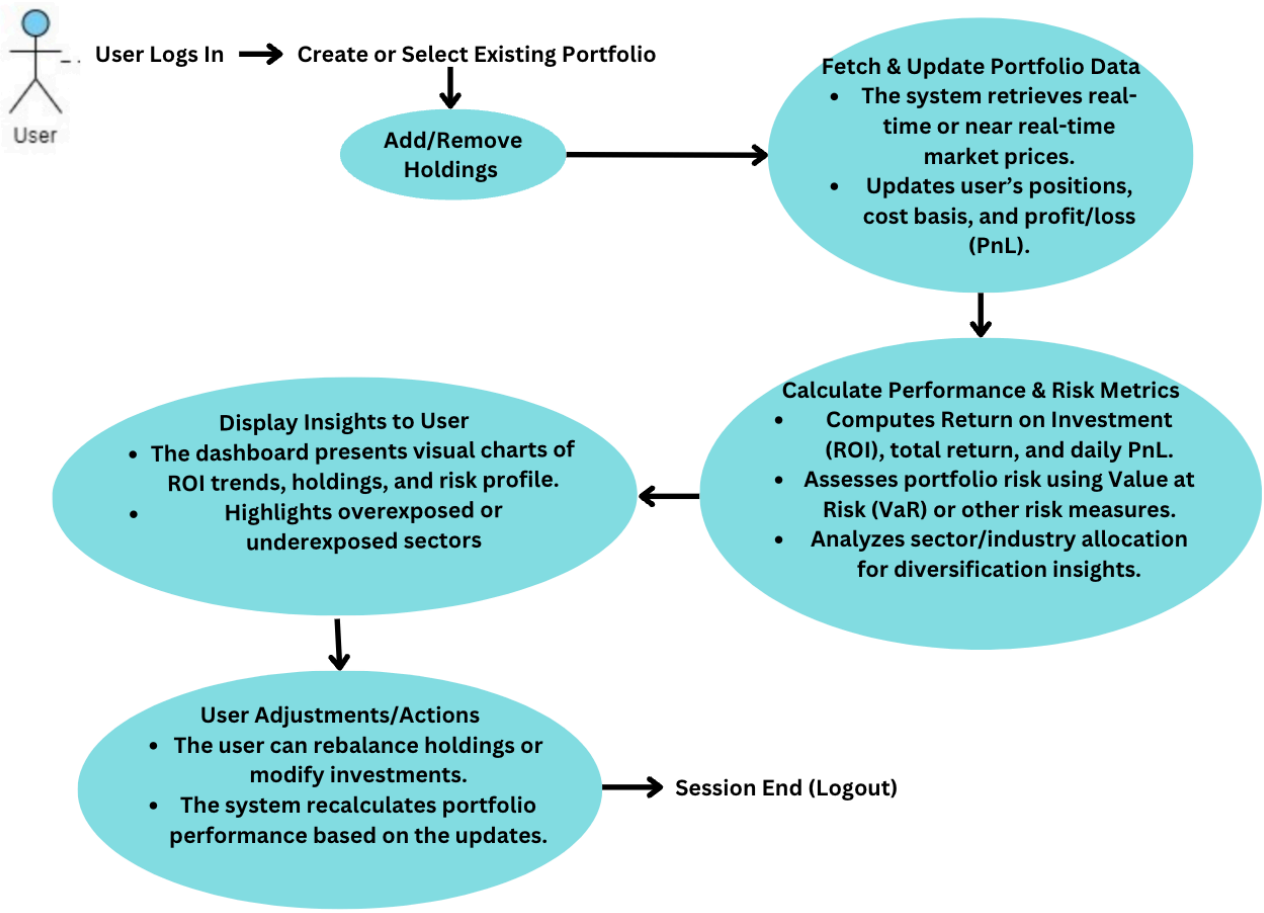
8. **Includes**

    - **User Authentication**
    - **Trading Simulation**
    - **Learning Content Management**

9. **Notes/Issues**

    - Decide reward amount for each activity.
    - Define any daily or weekly participation limits.
    - Keep educational content updated and engaging.

3.3.6. Use Case #6 Portfolio Management Tools:



Use Case: Portfolio Management

Authors: Aadi and Aruz

1. Purpose

Enable users to **create, view, and manage** their simulated portfolios, monitor **performance** (ROI, PnL), and evaluate **risk** (e.g., VaR, diversification) in real-time or near-real-time. This provides a comprehensive view of investments to aid in **informed decision-making** and **practice** within a virtual trading environment.

2. Requirements Traceability

- **REQ-PM-1**: Users must be able to **create and name multiple portfolios**.
- **REQ-PM-2**: The system must retrieve **real-time or near real-time market data** for accurate valuations.
- **REQ-PM-3**: Must provide **performance metrics** such as ROI, total and daily PnL.

- **REQ-PM-4**: Must compute **risk assessments**, including Value at Risk (VaR) and sector/industry breakdown.
- **REQ-PM-5**: Users must be able to **add or remove holdings** (buy/sell or import existing holdings).
- **REQ-PM-6**: Present **visual dashboards** (charts, tables) summarizing portfolio performance and risk.

3. Priority

Medium – Core functionality for a simulated trading platform. Accurate and user-friendly portfolio management is essential for user engagement and learning.

4. Preconditions

1. The user is **logged in** (or has registered and authenticated).
2. The system has **access to market data** (latest prices, volumes, etc.).
3. The user's portfolio exists or the user is able to **create a new one**.
4. The user's **virtual balance** is sufficient if attempting to add new holdings.

5. Post Conditions

1. **Portfolio holdings** (stocks, etc.) are updated with the latest transactions.
2. **Performance metrics** (ROI, PnL) and **risk analyses** (VaR, diversification) are recalculated.
3. A **dashboard** or **summary** view is presented to the user with real-time updates.
4. **Transaction records** (e.g., buy, sell) are stored for future reference.

6. Actors

- **User**: Initiates actions such as creating a portfolio, adding/removing holdings, and reviewing performance.
- **System**: Fetches market data, calculates performance/risk metrics, and updates portfolio information.

7. Exceptions

1. **Data Retrieval Failure**: Market data not available or delayed, causing incomplete or stale portfolio valuations.
2. **Insufficient Balance**: User attempts to purchase shares without enough virtual funds.
3. **Invalid Input**: Quantity, price, or symbol errors lead to rejection of the transaction.
4. **Session Timeout**: User is logged out before saving changes or viewing results.

8. Includes (Other Use Case IDs)

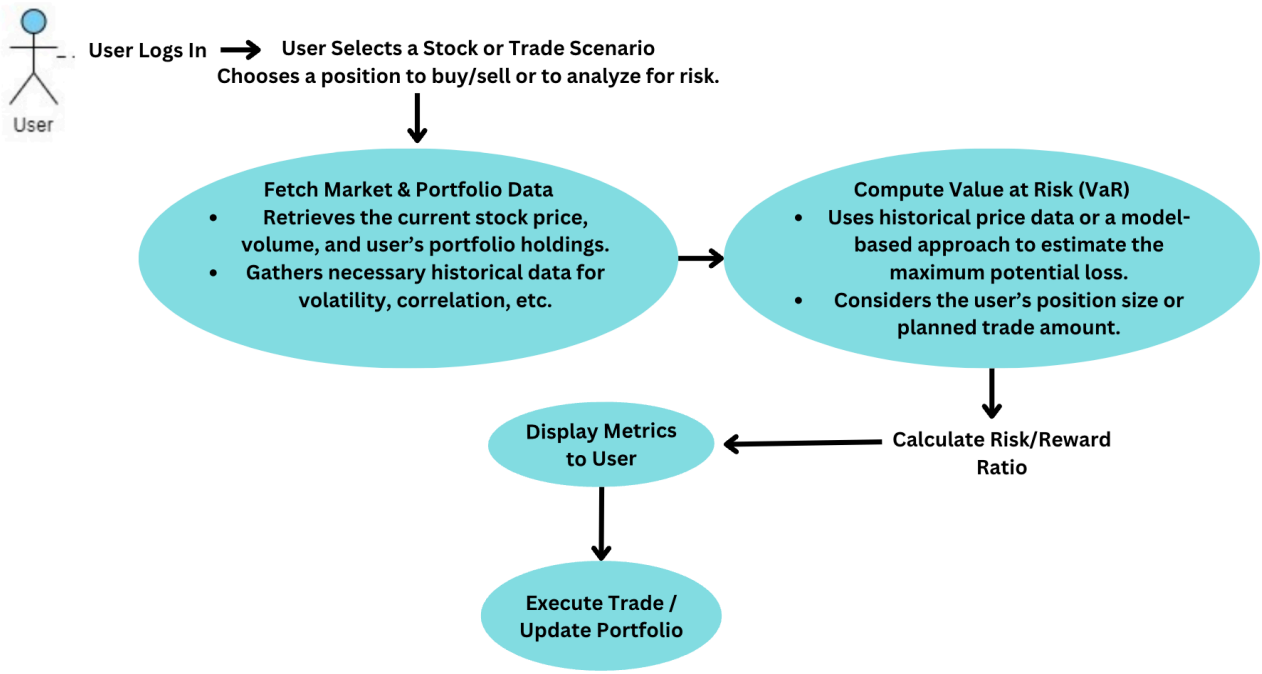
- **User Authentication** (for login/registration).
- **Risk Analysis** (for VaR, risk/reward calculations).

- **Trade Simulation** (where actual buy/sell mechanics are processed).

9. Notes/Issues

- Decide whether **partial shares** are permitted.
- Validate how frequently **risk metrics** (e.g., VaR) should be updated—real-time vs. daily batch.
- Ensure the system handles **large portfolios** efficiently and remains responsive.
- Confirm how the system will **log historical transactions** for review and analytics.

3.3.7. Use Case #7 Risk Analysis Tools:



Use Case: Risk Analysis

Authors: Aadi and Aadya

1. Purpose

To provide users with **quantitative metrics** (e.g., Value at Risk, risk/reward ratio) and **insight** into the potential losses or gains of their trades or portfolio. This helps them make **informed decisions** by understanding the risks before executing trades.

2. Requirements Traceability

- **REQ-1**: Calculate and display Value at Risk (VaR) for user portfolios.
- **REQ-2**: Provide a risk/reward ratio for each potential trade.
- **REQ-3**: Retrieve real-time or near-real-time market data to ensure accurate risk calculations.
- **REQ-4**: Update and present the metrics in a user-friendly dashboard or screen.

3. Priority

Medium – Accurate and timely risk analysis is crucial for the platform's educational and practical value. Users rely on these metrics to avoid excessive losses and understand potential rewards.

4. Preconditions

- The user has **portfolio data or a potential trade scenario** to analyze.
- The system can **access real-time/near real-time market data**.
- Historical data for **price volatility** or **model-based approach** is available for VaR calculations.

5. Post Conditions

- The user is presented with **VaR** (e.g., "95% chance of not losing more than ₹X") for their portfolio or selected trade.
- The system shows a **risk/reward ratio** (e.g., 1:2) to help the user gauge potential profits vs. potential losses.
- The user may **adjust or confirm trades** based on these risk metrics.

6. Actors

- **User**: Initiates a risk assessment for a specific trade or overall portfolio.
- **System**: Performs calculations (VaR, risk/reward) and returns results to the user.

7. Exceptions

1. **Data Unavailability**: Live or historical data feed fails, making risk calculation inaccurate or impossible.
2. **User Has No Holdings**: If the user's portfolio is empty, the system cannot compute VaR for that portfolio.
3. **Incorrect Inputs**: If user inputs an invalid trade size or incomplete details, the system cannot calculate risk metrics.
4. **Session Timeout**: If the user is logged out before calculations complete.

8. Includes (Other Use Case IDs)

- **User Authentication** (login/registration).
- **Portfolio Management** (to retrieve user holdings).
- **Market Data Retrieval** (to gather live/historical price data).

9. Notes/Issues

- Need to determine **which VaR method** (historical simulation, parametric, or Monte Carlo) to use.
- Clarify **frequency of updates** to risk metrics (on-demand or periodic).
- Ensure **performance** when handling large portfolios or real-time data bursts.

4. Other Non-functional Requirements

4.1. Performance Requirements

- A scalable architecture should be implemented to handle spikes in traffic, such as during mock trading events or peak hours
- The system should handle at least 100 transactions per second (TPS), including buy/sell orders, balance updates, and portfolio modifications.
- The platform should be horizontally scalable to accommodate an increase in user base or transaction volume by at least 100% without a complete overhaul of the architecture.
- API endpoints for critical operations, such as fetching stock data, must have a latency of less than 200ms under standard load conditions.
- The system must support real-time updates for stock prices, and order statuses with a latency of less than 500ms for data pushed to clients.
- The system should avoid race conditions and ensure transaction atomicity, even under high load. Updates to user portfolios and balances must maintain strong consistency and complete within 1 second of a transaction being executed.

4.2. Safety and Security Requirements

- The software ensures secure user authentication during login by enforcing the use of a strong password. A strong password must include a combination of uppercase and lowercase letters, numbers, and special symbols. Passwords are securely stored using hashing and salting techniques to protect against unauthorized access. Similarly, one-time passwords (OTPs) are also hashed before being stored in the database, ensuring the integrity and confidentiality of user authentication data.
- All connections to the server should use Transport Layer Security (TLS 1.2/1.3) encryption. All transaction data should be encrypted properly. The data at rest in the cloud should also be protected using advanced encryption techniques.
- Frequent backups should be made so that when some issue occurs, the user can get back his data as it was in the near past.
- The administration should delete all data related to a user upon the user’s request or after 6 months after a user terminates his account.

4.3. Software Quality Attributes

4.3.1 Ease of use and learning:

The software is designed to be user-friendly, enabling users to learn and operate it effectively within two hours of training or by referencing a maximum of ten help screens. Within two days of usage, the frequency of errors made by users is expected to drop to less than 1 in 100 transactions. This is particularly critical as the software is tailored for retailers, many of whom may have limited technical experience. To achieve this, the platform features a clean and intuitive user interface, with only essential options displayed and no unnecessary information clutter, ensuring a seamless and efficient user experience.

4.3.2 Maintainability:

The software will be designed with a modular and organized architecture, ensuring that new features can be added or existing ones modified with ease. Small modifications to existing features or the addition of minor features should take no more than two working days. This will be achieved by maintaining a clear separation between the various components of the software. Additionally, all code will be thoroughly documented with proper comments to enhance readability. To further improve efficiency and minimize errors, the team will adopt pair programming practices during development.

4.3.3 Reliability:

The software will be designed to efficiently handle high concurrent user traffic, ensuring reliable and uninterrupted access even during peak periods. It will seamlessly support an average load of 20,000 to 50,000 requests per day, maintaining high responsiveness and minimal downtime to ensure smooth execution of critical financial operations for users.

4.3.4 Portability:

As a web-based application, the software is accessible from any device with browser support. The frontend is built using Next.js, ensuring it is responsive and functions as a progressive web app, providing seamless performance across various platforms.

Appendix A – Data Dictionary

User Database

Variable	Description	Requirements
id	Unique ID of the user	Unique for each user
name	Name of the user	String
email	Email of the user	String, must be unique, valid email format
dob	Date of birth	Date
phone_no	Phone number of the user	String, must be unique
password	Password of the user	Hashed string (bcrypt)
balance	Balance in the user's account	Float
is_verified	Whether the user is verified	Boolean
created_at	Date and time the user was created	Timestamp (auto-generated)
updated_at	Date and time of last update	Timestamp (auto-updated)

Ticker Database

Variable	Description	Requirements
id	Unique ID of the company	Unique for each company
name	Name of the company	String
symbol	Stock ticker symbol	String, must be unique
industry	Industry the company belongs to	String

Transaction Database

Variable	Description	Requirements
id	Transaction ID	Unique for each transaction
user_id	ID of the user who made the transaction	Foreign key to User Database
stock_id	ID of the stock bought or sold	Foreign key to Ticker Database
volume	Number of stocks bought or sold	Integer (non-negative)
timestamp	Date and time of the transaction	Timestamp (auto-generated)
stock_price	Price per stock at the time of transaction	Decimal/Float
transaction_type	Type of transaction (buy/sell)	Enum (`BUY`, `SELL`)

OTP Database

Variable	Description	Requirements
user_id	ID of the user	Foreign key to User Database
otp	One-time password generated	String (hashed for security)
expiry_time	Expiration time of the OTP	Timestamp
created_at	Date and time the OTP was created	Timestamp (auto-generated)

Appendix B - Group Log

From the start of the project, our entire team has been highly enthusiastic. To ensure effective communication, we have created a WhatsApp group.

Date	Timing	Duration	Agenda
11/01/25	12pm- 2pm	2hrs	Brainstormed various ideas for the project and filtered 3-4 ideas out of them.
12/01/25	6pm-730pm	1.5hrs	Finalized the idea for the project and discussed various aspects of it.
15/01/25	9pm-12pm	3hrs	Discussed various features for the software and its feasibility. Decided the project name. Decided to use PostgreSQL, Golang and next.js
18/01/25	3pm-430pm	2hrs	Studied the SRS template given and distributed the work amongst the team members.
20/01/25	6pm-8pm	2hrs	Brainstorming of final ideas and discussion on use cases, features, dataflows. The team was divided into groups to think about the use cases.
22/01/25	9pm-1130pm	2.5hrs	Finalized the use cases, Product Overview, Functional Requirements and Non- Functional requirements for the project. Reviewed the SRS after compiling the parts made by each team.
24/01/25	4pm-430pm	0.5hrs	Google Meet with Teaching Assistant Mr. Naman Baranwal. Discussed about the project and the SRS document.
24/01/25	5pm-7pm 9pm-1130pm	4.5 hrs	Finalized the SRS document