REQUIREMENT ANALYSIS DOCUMENT

Problem Statement: Education-Oriented Financial & Paper Trading Platform

Submitted By:

Jeswanth Reddy - S20240010202
Purvaj P - S20240010178
Bharan Raj P - S20240010176
Gurwinder Singh - S20240010082
Mohan Sai Praneeth P - S20240010179

Course: B.TECH - Computer Science & Engineering
Institution: IIIT SRICITY
Project: FFSD PROJECT
Faculty Guide: Dr. Anushree Bablani
Academic Year: 2026-2027

Introduction

Purpose:

The purpose of this document is to define the functional and non-functional requirements of the *Education-Oriented Financial Literacy and Paper Trading Platform*. This document serves as a reference for developers, instructors, and evaluators to understand system behavior, constraints, and expected features.

Scope of the project:

The system is a web-based platform that enables students to learn basic financial concepts and practice stock trading using virtual money. It includes learning modules, paper trading simulation, portfolio tracking, instructor dashboards, and administrative tools.

Definitions:

Term	Meaning
Paper Trading	Simulated trading using virtual money
Portfolio	Collection of virtual assets held by a user
Admin	System administrator
Instructor	Faculty user monitoring students
Trading Assignment	A practical trading task assigned to student traders for skill development
Enrollment	The association of a student trader with a course
Market Price Feed	Source of financial instrument price information used in trading simulation
Course Module	A learning unit containing educational content, exercises, and assessments
Financial Instrument	A tradable market asset such as stocks, ETFs, or indices available in simulation

Term	Meaning
Matching Engine	Core component responsible for matching buy and sell orders based on price and time priority
Order Book	Data structure holding all pending buy and sell orders for a stock
Market Order	Order that executes immediately at the best available price
Limit Order	Order that executes only at a specified price or better
Stop-Loss Order	Order triggered when the market price reaches a predefined threshold
Partial Fill	When only part of an order quantity is matched
Cash Check	Verification that a user has enough balance before placing an order
Margin Check	Verification that margin requirements are satisfied (if margin trading is enabled)
Order Book Snapshot	Periodic saved state of the order book for recovery and visualization
Asynchronous Database Writes	Writing trade and portfolio updates to the database in the background
Latency	Delay in market data updates, considered non-critical for learning systems

DESCRIPTION

Product Perspective:

The platform is a standalone web application developed using a full-stack architecture. It integrates a frontend user interface, a backend server for business logic, and a database for persistent storage. Market price data may be fetched from a third-party API or a static dataset.

USER CLASSES:

User Type	Description
Student	Learn Financial concepts and performs paper trading
Instructor	Assigns tasks, monitors student performance
Course provider	Provide courses and create assignments
Admin	Manages users, learning content, and stock data

Operating Environment:

The system will work on common web browsers like Chrome, Firefox, and Edge, so users don’t need to install anything special.The main program will run on a computer or an online server that handles all the background work.All the user data and app information will be stored safely in a database system.

User Requirements

Student Requirements:

→Students should be able to create an account and log in securely.

→Students should be able to view real-time or near real-time stock prices.

→Students should be able to buy and sell stocks using virtual money.

→Students should be able to view their virtual portfolio and transaction history.

→Students should be able to track profit/loss and overall performance.

→Students should be able to access basic financial learning modules and guidelines.

Instructor Requirements:

→Instructors should be able to log in securely.

→Instructors should be able to view student performance and portfolios.

→Instructors should be able to assign tasks or trading challenges.

→Instructors should be able to monitor learning progress.

→Instructors should be able to provide feedback or remarks.

Admin Requirements:

→Admin should be able to manage users (add, delete, block, update).

→Admin should be able to manage stock data sources and APIs.

→Admin should be able to manage learning content.

→The admin shall be able to configure trading rules such as virtual capital limits, market hours, allowed instruments, and risk constraints.

→The Admin shall be able to enable, disable, or restrict platform features based on academic or operational requirements.

Course Provider Requirements:

→The Course Provider shall be able to create and submit course offerings including course title, description, learning objectives, duration, and prerequisites.

→The Course Provider shall be able to update or revise course content based on market changes, learner feedback, or academic requirements.

→The Course Provider shall be able to publish updates or announcements related to course content or schedule changes.

→The Course Provider shall be able to manage assessments and evaluation components associated with each course.

→The Course Provider shall be able to define scoring and ranking mechanisms for trading challenges.

→The Course Provider shall be able to integrate challenge results into course assessments.

SYSTEM REQUIREMENTS

→User authentication and role-based access control (Student / Instructor / Admin).

→The system should automatically update stock prices and the daily report of each user’s paper trading portfolio.

→Users should be able to view real-time or near real-time stock data and search stocks by name, symbol, or sector.

→Paper trading engine with virtual money and virtual portfolios for each user.

→Trade execution system to simulate buy/sell operations with balance updates.

→Automatic transaction logging for every trade (date, time, price, quantity, profit/loss).

→Instructor monitoring dashboard to track student performance, trades, and learning progress.

→Portfolio performance analytics including returns, risk indicators, and trade history.

→Learning module integration for financial concepts and trading rules.

→Notification system for trade confirmations, portfolio updates, and system messages.

→Data storage system for user profiles, portfolios, transactions, and learning progress.

→Secure backup and recovery mechanism to prevent data loss.

Functional Requirements

Authentication:

→The system shall allow users to register and log in.

→The system shall validate user credentials.

Trading Simulation:

→The system shall display stock prices.

→The system shall allow buying and selling stocks using virtual money.

→The system shall update the user portfolio in real time.

Portfolio Management:

→The system shall display portfolio value.

→The system shall calculate profit/loss.

→The system shall store transaction history.

Learning Module:

→The system shall display educational content.

→The system shall allow instructors to add learning content.

→The system shall track content completion.

Admin Management:

→The system shall allow admin to manage users.

→The system shall log system activities.

Non-Functional Requirements

Performance:

→The system should respond within 2 seconds for user actions.

→The system should handle at least 500 concurrent users.

Security:

→The system should encrypt user passwords.

→The system should use HTTPS.

→The system should prevent unauthorized access.

Usability:

→The system should be easy to use for beginners.

→The system should support both desktop and mobile browsers.

Reliability:

→The system should have at least 99% uptime.

→The system should handle failures gracefully.

Scalability:

→The system should support future feature expansion.

→The system should handle increased user load.

UML DIAGRAMS

Use case diagram : UML diagram of use cases

Activity Diagrams : UML diagrams for activities

Features Diagram : UML diagram of features

Sequence Diagram :UML diagram of actor sequences