Simulation Game for Learning Algorithmic Trading

Cheung Sui Wing, 21027547D

*Abstract*— The Simulation Trading Game System is a web-based platform designed to educate individuals about algorithmic trading and enable them to test their trading strategies in a risk-free environment. The system comprises a frontend and backend architecture, a RESTful server, and a MongoDB document database. The system provides users with three algorithm levels to explore different trading strategies and learn about risk management.

# Introduction

Algorithm trading can be complex, making it challenging for new entrants. The simulations game enables interactive learning and provides users with hands-on experience executing trading strategies using market data and simulated trades. Users can safely test and improve their skills with step-by-step tutorials in a risk-free environment to enhance understanding. The proposed game system benefits beginners, providing an immersive and interactive experience that enables the development of skills and knowledge in algorithm trading.

# Overview

The simulation trading game is a web-based platform developed using React, Node.js, Express.js, and MongoDB Atlas. It utilizes historical data from polygon.io to give users realistic and diverse market conditions for testing their trading strategies. The game offers three levels of complexity, enabling users to experiment with different trading strategies and develop their skills through step-by-step guides, tooltips, and other forms of guidance. The game also includes a history feature and leaderboard, providing users with a detailed view of their trading performance and a fun and engaging way to compete with other users. Overall, the simulation trading game offers a secure, scalable, and interactive learning experience for users interested in algorithmic trading.

# System architecture

The system architecture of the simulation trading game consists of a front end built using React, a RESTful server as the back end, and various third-party components. The system retrieves historical data using HTTP requests and stores user data in a MongoDB document database. JSON Web Tokens (JWTs) are used to protect routes and ensure system security. The frontend and backend communicate via RESTful APIs, and the backend interacts with the database to retrieve and store data. The security component verifies user identity when accessing protected routes. The main flow of the system includes the user requesting a new game, selecting assets, configuring rules, and the server performing simulation tasks and saving records to the database.

# Algorithms Provided:

Three algorithms are provided in the game: Dollar Cost Averaging (DCA), Martingale, and Custom Rules. The DCA algorithm involves investing a fixed amount of money at regular intervals, regardless of market conditions, to reduce the impact of market volatility. The Martingale algorithm is a high-risk betting strategy that involves increasing the amount of the next bet after a loss so that the first win would recover all previous losses plus a profit equal to the original stake. The Custom Rules algorithm provides users with various rule-setting options, including technical indicators, to optimize their trading strategy.

The Custom Rules feature allows users to set up rules based on technical indicators like Moving Average, RSI, and MACD. Users can create groups of buy/sell conditions with different operation types, including And, Not, and Count. Each rule consists of three components: Expression1, Operator, and Expression2. The system processes all rules within a group based on the selected operation type and generates a final result, executing the buy/sell order if all rules pass.

Diagram

Description automatically generated

Figure 1 Structure of the condition groups

# Integration with Course in PolyU

The software developed in this paper has the potential to be utilized in courses related to finance, technology, and algorithmic trading at The Hong Kong Polytechnic University. The simulation game can provide students practical experience in trading technology and algorithmic trading, allowing them to test different trading strategies and analyze the results. The software can supplement the theoretical concepts taught in courses such as COMP4141 Crowdfunding and E-Finance and COMP4531 Emerging Topics in FinTech.

# Conclusion

This paper describes the development of a simulation trading game system that provides users with an interactive and educational tool to learn about algorithmic trading and test their trading strategies in a risk-free environment. The system's architecture, frontend and backend, and algorithm design have been detailed, along with the API routes and testing procedures. The system provides three algorithm levels, enabling users to explore different trading strategies and learn about risk management. Overall, the system provides a valuable tool for individuals interested in algorithmic trading to learn and develop their trading skills.