The University of Hong Kong
Department of Computer Science

2024–2025 COMP4801 Final Year Project

Interim Report

A Web Solution for Backtesting and Optimizing Retail Futures Trading

Supervisor:	Dr. Kam Pui Chow	
Members:	Kwong Chiu Lam	3035927427
	Wong Yee Wang	3035788637

Date of Submission: 26th January, 2025

ABSTRACT

This project is a web-based solution for backtesting and optimizing futures trading strategies, allowing retail trader to evaluate and enhance their strategies with historical market data. The website provides three backtesting methods: Default Strategies, Replay Backtesting, and Portfolio Backtesting, all integrated with intuitive visuals and performance metrics. This platform is developed on a three-tier architecture: Presentation Layer, Application Layer, and Data Layer. Each layer utilizes specific tools and technologies to guarantee a smooth user experience and optimal operation. Future improvements may include strategies suggestions by machine learning and the creation of mobile version.

ACKNOWLEDGEMENTS

We would first like to express my sincerest gratitude to our supervisor, Dr. Kam Pui Chow, for his continuous guidance and support throughout this project. His expertise and feedback have been crucial in the success of this work. I would also like to express our appreciation to the Department of Computer Science at The University of Hong Kong for providing the resources to complete this project. The knowledge and skills I have learned during my studies have also been imperative in the development of this web solution.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. Introduction

Recently, there has been an upsurge in retail participation in financial markets, be it in traditional equity trading or the derivatives market, such as futures trading. In 2021, retail investors represented over 47.4% of the total global securities market volume, a rise from 39.6% in 2017 and is expected to represent 48.8% in 2026 [1]. Technological advancements are a pushing factor in this trend. Online brokers and mobile apps like Webull, and Futu have made trading more accessible. Simultaneously, these online brokerages often offer low or even zero trading fees with no capital requirements, thereby encouraging more retail market involvement. Furthermore, the COVID 19 pandemic enabled individuals to spend more time at home, leading some to engage in trading as a substitute for gambling [2]. The retail trading market is expected to continue its growth.

Futures trading is characterised by high liquidity and volume, which are critical for accurate backtesting and strategy development. The Futures Industry Association (FIA) reported that overall trading volume in 2023 reached 137.3 billion contracts, a 64% increase from the previous year. This is the sixth consecutive year of record-setting activity in global listed derivatives markets [3]. Futures contracts also allow the control of significant positions with very small capital investment, using leverage of approximately 20 times, in contrast to stock trading, which commonly provides typically 2 times leverage. This leverage renders futures trading particularly attractive to retail traders aiming to maximize their investment potential. Additionally, futures encompass a variety of asset classes, including commodities, indices, currency and interest rates, enabling investors to diversify their portfolios efficiently.

The remaining parts of this report are organized as follows: First, we outline the project background, motivations, and objectives in Section 1. Then, we discuss the methodologies for constructing the web solution in Section 2. Next, we present the current progress and future development plan in Section 3. Finally, we conclude by summarizing the findings in Section 4.

1.1. Motivations

1.1.1. Valuable Market

The global derivatives market value reached USD 137.3 billion in 2023 (approximately HKD 1,071 billion) [3]. Retail traders will comprise 48.6% of overall market participants by 2026 [1]. Consequently, we anticipate continued growth in retail involvement in futures trading.

1.1.2. Knowledge Gap Among Retail Traders

Derivatives, especially futures, are complicated financial products associated with significant risks and returns. To prevent retail traders from obsessed with temporary gains while disregarding risks, it is essential to advocate backtesting techniques. This enables traders to practice and fine-tune their strategies prior to entering the real market.

1.1.3. Limitations in Current Platforms

Many online brokerages offer backtesting tools for retail traders; however, these tools often lack comprehensive capabilities. TradingView, one of the most widely used platforms for retail web-based backtesting with over 50 million active users globally (as of 2023), also has its limitations. Its user interface can be complex, and it typically does not support portfolio-wide testing. A detailed case study comparing TradingView and our solution is provided in Appendix A.

1.2. Project Aims and Objectives

This project aims to develop a web-based backtesting solution specifically for futures trading. The platform will allow individuals without any programming knowledge to evaluate their trading strategies utilizing historical market data from the Chicago Mercantile Exchange (CME) in an intuitive user interface, providing comprehensive performance metrics and interactive visualizations.

Considering the continuous growth of retail participation in the futures market, the knowledge gap among retail traders, and the limitations of current platforms, this project strive to achieve the following objectives:

1. To provide a user-friendly platform that streamlines the backtesting process for users without any programming experiences.
2. To educate users on risk management and strategy development through interactive tools.
3. To offer comprehensive portfolio testing features, enabling users to evaluate multiple assets and optimize diversification.

2. Methodologies

Backtesting is a core component of this project, enabling users to evaluate the performance of trading strategies through simulations using historical market data. This process allows users to refine their strategies, minimize risks, and make data-driven decisions before engaging in live trading. Our platform offers two primary categories of backtesting methods: Automated Backtesting and Manual Backtesting, each designed to meet the diverse needs and proficiency levels of traders. Both approaches integrate performance metrics to help users analyze the efficacy and risks of their strategies.

2.1. Automated Backtesting

Automated backtesting allows users to evaluate trading strategies systematically by leveraging predefined rules, algorithms, and statistical techniques. It minimizes manual intervention and provides a structured environment for comprehensive strategy evaluation. This category includes the following methods:

2.1.1. Default Strategies with Parameter Modification

The Default Strategies with Parameter Modification method is designed for users with little to no experience in futures trading. It offers predefined strategies, such as the RSI (Relative Strength Index) Strategy, which identifies overbought or oversold market conditions, and the BarUpDn Strategy, which uses price patterns to predict trends.

To simplify strategy implementation, the platform also provides optimized parameters for each default strategy. These optimized parameters are determined through historical data analysis and algorithmic optimization techniques, ensuring that users start with configurations that have historically demonstrated strong performance. The optimization process involves:

1. Historical Data Analysis: Testing the strategy across a wide range of historical market data to identify parameter settings that yield the best performance in various conditions.
2. Grid Search Optimization: Running exhaustive tests across a predefined range of parameter values (e.g., RSI thresholds, moving average periods) to find

combinations that maximize performance metrics such as profit, win rate, and reward-to-risk ratio.

3. Walk-Forward Analysis: Validating the optimized parameters by applying them to unseen historical data to ensure they generalize well across different market conditions and are not overfitted.
4. Performance Metrics Evaluation: Selecting the parameter set that demonstrates the best balance between profitability, risk management, and robustness, based on metrics such as Profit and Loss (P&L), Sharpe Ratio, and Maximum Drawdown.

Users are encouraged to use these optimized parameters as a starting point, while still having the flexibility to adjust them based on their individual preferences and market conditions.

Key parameters that can be adjusted include:

- RSI thresholds to define overbought and oversold levels,
- Moving averages to identify trend directions, and
- Stop-loss limits to manage risk.

To evaluate these strategies, the platform provides key performance metrics, including:

- Profit and Loss (P&L): Displays the overall net gain or loss.
- Win Rate: Reflects the proportion of profitable trades.
- Reward-to-Risk Ratio: Calculates the expected return relative to risk taken.

This method is ideal for beginners seeking a straightforward way to explore and customize trading strategies.

2.1.2. Portfolio Backtesting

The Portfolio Backtesting method focuses on evaluating the performance of an entire portfolio rather than individual trades. It is particularly suited for advanced users who aim to optimize diversification and manage risk across multiple assets.

This approach supports multi-asset backtesting across various futures contracts, including:

- Commodities (e.g., crude oil),
- Indices (e.g., Nasdaq Composite), and
- FX (Foreign Exchange) rates (e.g., EUR/USD futures).

By analyzing asset correlations, users can identify and avoid over-concentration in highly correlated assets, enabling them to achieve better diversification. Key evaluation indicators for portfolio backtesting include:

- Portfolio Return: Measures the overall returns across all assets.
- Sharpe Ratio: Compares portfolio returns with associated risks to gauge risk-adjusted performance.
- Maximum Drawdown: Identifies the largest decline in portfolio value during the testing period.

Portfolio backtesting is ideal for traders seeking to manage risk holistically and ensure the performance of diversified portfolios.

2.1.3. Monte Carlo Simulation

The Monte Carlo Simulation method uses statistical modeling to assess the robustness and potential performance of trading strategies under a wide range of market conditions. By introducing randomness and variability to key factors, this method generates thousands of possible outcomes, allowing traders to evaluate the effects of uncertainty on their strategies. Key features of Monte Carlo Simulation include:

- Randomized Trade Variables: Simulates variations in trade outcomes, such as sequence, price fluctuations, slippage, and market volatility.
- Stress Testing: Assesses the performance of strategies under extreme market conditions, such as sudden price shocks or prolonged drawdowns.
- Outcome Distribution: Provides a range of possible performance metrics, including best-case, worst-case, and most likely scenarios.

Performance metrics used in Monte Carlo Simulation include:

- Expected Return: The average return across all simulated outcomes.
- Maximum Drawdown: The largest loss observed in any simulation.
- Win Probability: The likelihood of achieving specific performance thresholds.

Monte Carlo Simulation is particularly valuable for advanced users who want to understand the variability, risks, and potential vulnerabilities in their strategies. By providing a probabilistic view of potential outcomes, this method helps traders make more informed decisions and better manage risk.

2.2. Manual Backtesting

2.2.1. Replay Backtesting

Replay Backtesting is a powerful tool that allows traders to simulate real-time market conditions using historical data. Unlike demo trading, which often relies on live market data, Replay Backtesting lets users "replay" past market scenarios, providing an interactive and immersive way to test trading strategies.

Key features of Replay Backtesting include:

- Historical Market Simulation: Users can reset the timeline to significant historical events, such as the 2008 Financial Crisis or the COVID-19 pandemic, to evaluate strategies in extreme market conditions by simulated "real-time" trading.
- Playback Speed Control: Users can slow down or speed up market movements to conduct in-depth analyses or rapid testing.
- Scenario Analysis: Traders can explore how their strategies perform under various historical events.

Crucial evaluation metrics for this approach include:

- Cumulative Profit: Tracks the total profit or loss accrued throughout the simulation period, giving a clear picture of overall performance.
- Equity Curve: Monitors fluctuations in account balance over time, helping users visualize their trading success and risk exposure.
- Trade Accuracy: Assesses the ratio of successful trades to total trades executed, providing insight into the effectiveness of the trading strategy.

Replay Backtesting is particularly useful for traders who want to manually refine their strategies and analyze market behavior in a controlled environment.

2.2.2. Journaling and Checklist

In manual backtesting, maintaining a Journal and using a Checklist are essential practices that help traders systematically evaluate their strategies and improve their performance. These tools serve to document decisions, reflect on outcomes, and ensure a thorough analysis of trading behavior.

Key components of trading journal include:

- Trade Documentation: Record details of each trade, including entry and exit points, position size, and the rationale behind the trade. This helps in analyzing what works and what doesn’t.
- Outcome Analysis: Evaluate the results of each trade, including profits or losses, and compare them against your initial expectations.
- Strategy Review: Regularly assess the effectiveness of strategies used by identifying patterns in successful and unsuccessful trades.

Key components of trading checklist include:

- Market Conditions: Assess current market conditions to determine if they align with your trading strategy.
- Setup Criteria: Confirm that all technical and fundamental criteria have been met before entering a trade (e.g., signals from indicators, support/resistance levels).
- Risk Management: Ensure that risk management rules are in place, such as stop-loss levels and position sizing.

Incorporating a Journal and Checklist into manual backtesting can significantly elevate a trader's approach. By systematically documenting trades and adhering to a structured process, traders can refine their strategies, improve decision-making, and ultimately enhance their trading performance.

3. System Design of the Website

Figure 1 illustrates that the system is constructed on a three-tier architecture: Presentation Layer, Application Layer, and Data Layer. Each layer utilizes specific tools and technologies to guarantee a smooth user experience and optimal operation.

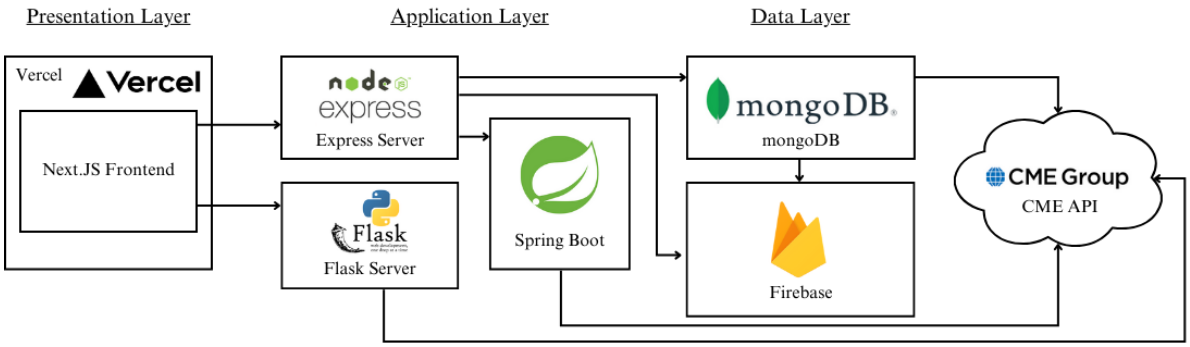

Figure 1: *The system flowchart of the website application which illustrated the communication between presentation, application, and data layers*

3.1. Presentation Layer

The web backtesting platform is developed using ReactJS with the Next.js framework to deliver a fast, responsive, and user-friendly interface [4]. This technology ensures smooth navigation and quick loading times, which are essential for accessing real-time trading data and interactive charts. To enhance performance and user experience, the platform employs a combination of Server-Side Rendering (SSR) for dynamic trading data and backtesting results, and Static Site Generation (SSG) for static content like tutorials and documentation. The frontend is hosted on Vercel, a serverless hosting platform that provides benefits such as edge caching, global content delivery, and a continuous integration/continuous deployment (CI/CD) pipeline [5]. This arrangement ensures reliability, scalability, and consistent performance across devices with varying hardware and internet speeds. By leveraging Vercel's capabilities, users enjoy a seamless experience regardless of their connection quality or device performance.

3.2. Application Layer

The application layer is designed using a microservices architecture to efficiently manage the platform's diverse functionalities. Each service is tailored to specific tasks, ensuring flexibility, scalability, and ease of maintenance:

- Express.js**:** Acts as the primary API gateway, handling frontend requests such as retrieving user account information, routing data, or executing trades. Its lightweight nature enables rapid development and iteration of core features [6].
- Flask**:** Powers the backtesting and analytical tools, utilizing Python's robust data-handling libraries. Flask is responsible for executing backtesting logic, processing

historical market data, and performing initial strategy evaluations, offering users insights into their trading performance [7].

- Spring Boot**:** Handles complex computational tasks, such as strategy optimization and advanced risk analysis. It is designed for efficiency and scalability, ensuring the platform can seamlessly handle increased user demand and computational complexity as the system grows.

These technologies work together to deliver a responsive and adaptable application layer, providing users with powerful tools to refine their strategies and manage their trading activities effectively.

3.3. Data Layer

The data layer is structured to efficiently handle various types of data, ensuring flexibility, scalability, and real-time responsiveness. It integrates the following components:

- MongoDB: Stores unstructured data, such as trading history, backtesting results, and user preferences. Its NoSQL format allows for seamless adaptation to evolving customer demands and supports the scalability required for large datasets [8].
- Firebase: Enables real-time updates on platform events, such as notifications for completed backtesting tasks or immediate alerts on market changes. This ensures users can react promptly to shifting conditions, enhancing their trading experience [9].
- CME Group API: Provides access to real-time and historical market data, including OHLC (Open, High, Low, Close) data and trading volume. This data is critical for supporting features like strategy optimization, replay backtesting, and portfolio analysis. The backend caches this data in MongoDB for efficient retrieval, ensuring users have access to precise and timely information.

By combining these technologies, the data layer guarantees robust data management, allowing traders to make informed decisions based on accurate and up-to-date market insights.

4. Current Progress and Expected Results

Substantial progress has been achieved in the project, with most scheduled tasks completed as planned, while some tasks have faced minor delays (Table 1). In September, preparation activities such as the literature review, requirements analysis, and project planning were successfully completed. By October, the system design was finalized, defining the platform's architecture and selecting its core technologies. Additionally, the project plan was delivered on schedule.

However, delays occurred in November during the UI/UX design and frontend development stages. While the UI/UX design has been completed, unexpected challenges in aligning the design with user requirements caused minor setbacks. Despite this, the core frontend framework, utilizing ReactJS and Next.js, has been successfully established. The integration of responsive features into the UI took more time than anticipated, slightly altering the timeframe for finalizing the frontend architecture. To address these delays, additional efforts were made in December to streamline progress.

By December and January, significant strides have been made in backend development and data integration:

- The backend setup using Flask and Express.js has been successfully completed, laying the foundation for core functionality.
- Data integration efforts, including connecting to Firebase for real-time updates and sourcing market data from the CME Group API, have been implemented successfully.
- The Default Strategies Implementation (RSI Strategy and BarUpDn Strategy) is currently in progress, with parameter modification capabilities under development.

Despite the earlier delays, the project remains aligned with its overarching goals due to the robustness of the system architecture. The next steps involve finalizing the **default strategies** and implementing the performance metrics to evaluate strategy efficacy. Subsequent activities will include the development of Replay Backtesting and Portfolio Backtesting in February, followed by visualization and optimization tasks in March.

(Table 1 provides a detailed breakdown of completed and upcoming tasks)

Date	Task
September	- Literature Review (Done) - Requirements Analysis (Done) - Project Planning (Done)
October	- Deliverable 1: Project Plan (Done) - System Design (Done)
November	- Design of User Interface (UI) and User Experience (UX) (Done)
December & January	- Core Frontend Setup: Set up ReactJS and Next.js environment, Develop basic UI/UX framework (Done) - Backend Setup: Set up Flask and Express servers, Develop initial API structure (Done) - Data Integration: Connect to Firebase/AWS DynamoDB, Source CME data and ensure accessibility (Done) - Default Strategies Implementation: Develop RSI Strategy and BarUpDn Strategy, allow parameter modification, Implement performance metrics (In progress) - Deliverable 2: Interim Report
February	- Replay Backtesting Development: Enable timeline reset for historical trading - Portfolio Backtesting Features: Implement multi-asset testing
March	- Visualization Development: Use Plotly.js for interactive charts - Refinement and Optimization: Address feedback and improve usability, Optimize system performance - Comprehensive Testing: Conduct unit and integration testing, perform user testing and gather feedback
April	- Documentation and Finalization: Complete user manuals and technical documentation - Deliverable 3: Final Report - Deliverable 4: Promotional Video and Poster

Table 1: *Expected Deliverables & Project Timeline*

5. Conclusion and Future Works

In conclusion, this project aims to deliver a web-based platform designed to streamline retail futures trading backtesting and optimization. The website is expected to empower retail traders to evaluate and refine their strategies effectively through key features like Default Strategies, Replay Backtesting, and Portfolio Backtesting, all without requiring programming expertise. While significant progress has been made in system design, backend setup, and data integration, minor delays in UI/UX design and frontend development have required adjustments to the timeline. Moving forward, additional efforts will be dedicated to completing the remaining tasks, ensuring that the platform meets its objectives and provides a robust, user-friendly solution for retail traders.

Future work beyond the present project scope may involve exploring various additions that improve the platform's usability and usefulness. This includes the integration of machine learning models to suggest ideal trading strategies based on historical data and user behavior, the development of a mobile application to enhance accessibility, and the incorporation of real-time notifications and warnings for market fluctuations. The platform might also offer multi-language interfaces to accommodate a worldwide audience and develop to incorporate more asset classes such as stocks and options for enhanced market coverage.

These prospective initiatives would improve the platform's scalability, accessibility, and usefulness, ensuring it remains aligned with the changing requirements of retail traders.

6. References

[1] INDUSTRY OVERVIEW, 22-Dec-2022 [Online]. Available: https://www1.hkexnews.hk/listedco/listconews/sehk/2022/1222/10549659/2022122200094.pdf. [Accessed: 26-Sep-2024].

[2] A. Orujov, “Trading as gambling during covid-19 lockdown,” SSRN Electronic Journal, 2023.

[3] “Global futures and options volume hits record 137 billion contracts in 2023,” FIA. [Online]. Available: https://www.fia.org/fia/articles/global-futures-and-options-volume-hits-record-137-billion-contracts-2023#:~:text=The%20total%20number%20of%20futures,4.8%25%20to%2012.6%20billion%20contracts. [Accessed: 26-Sep-2024].

[4] Vercel, “Getting Started | Next.js,” nextjs.org, 2024. [Online]. Available: https://nextjs.org/docs. [Accessed: 15-Oct-2024].

[5] “What Is Vercel & How Is It For Front-End Developers?,” Intuji, Feb. 21, 2022. [Online]. Available: https://intuji.com/what-is-vercel-is-it-the-right-platform-for-front-end-developers/. [Accessed: 15-Oct-2024].

[6] W. Grant, “Express: The Lightweight and Beginner-Friendly Node.js Framework,” Medium, Mar. 31, 2023. [Online]. Available: https://medium.com/@wesley.grant/express-the-lightweight-and-beginner-friendly-node-js-framework-52ad639e3b22. [Accessed: 15-Oct-2024].

[7] A. Saini, “An Easy introduction to Flask Framework for Beginners,” Analytics Vidhya, Oct. 20, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/flask-python/. [Accessed: 15-Oct-2024].

[8] GeeksForGeeks, “MongoDB: An introduction,” GeeksforGeeks, Nov. 04, 2015. [Online]. Available: https://www.geeksforgeeks.org/mongodb-an-introduction/. [Accessed: 15-Oct-2024].

[9] “Firebase Introduction - Javatpoint,” Javatpoint. [Online]. Available: https://www.javatpoint.com/firebase-introduction. [Accessed: 15-Oct-2024].

7. Appendices

7.1. Appendix A: Case Study Comparing Future Vision and TradingView

Feature	Future Vision	TradingView
Backtesting Methods	Default, Replay, Portfolio backtesting	Pine Script, Replay (single asset)
Ease of Use	No coding required, intuitive UI	Requires Pine Script for custom tools
Market Focus	Specialized in futures trading	Broad market coverage
Performance Metrics	Advanced (e.g., Sharpe Ratio, Max Drawdown)	Basic (single asset focus)
Visualization	Interactive with Plotly.js (planned)	Advanced, real-time charting
Automation	Predefined strategies, ML integration (future)	Pine Script automation
Replay Mode	Historical event simulation, adjustable speed	Basic historical replay

Table 2: *Summary table comparing Future Vision & TradingView*

7.2. Appendix B: UI/UX Design

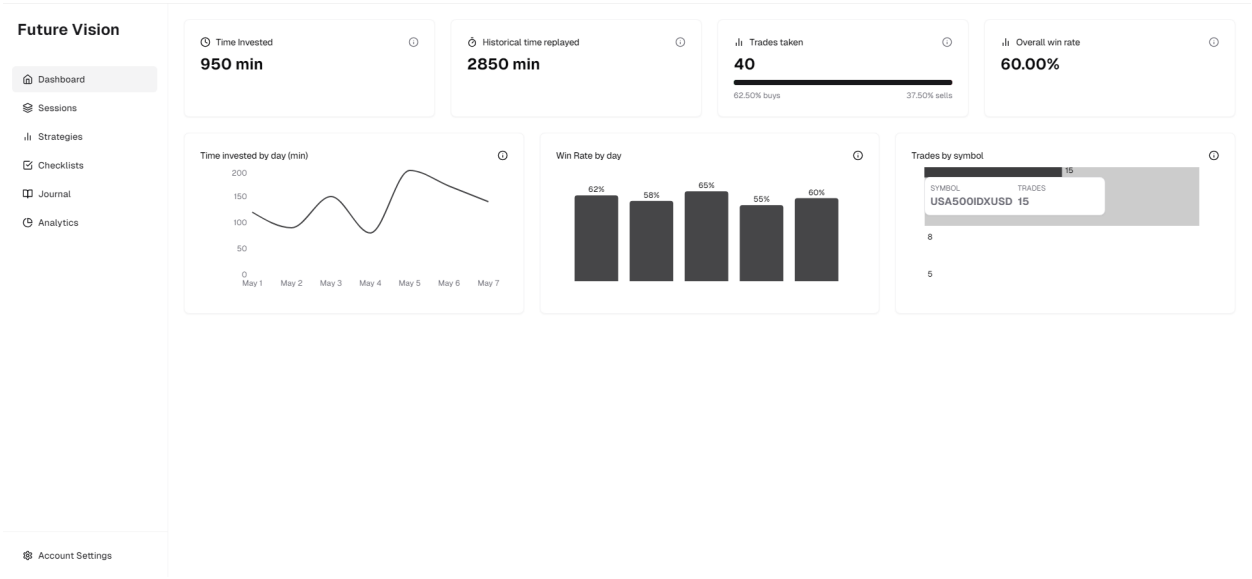

Figure 2: *Dashboard Page of Future Vision*

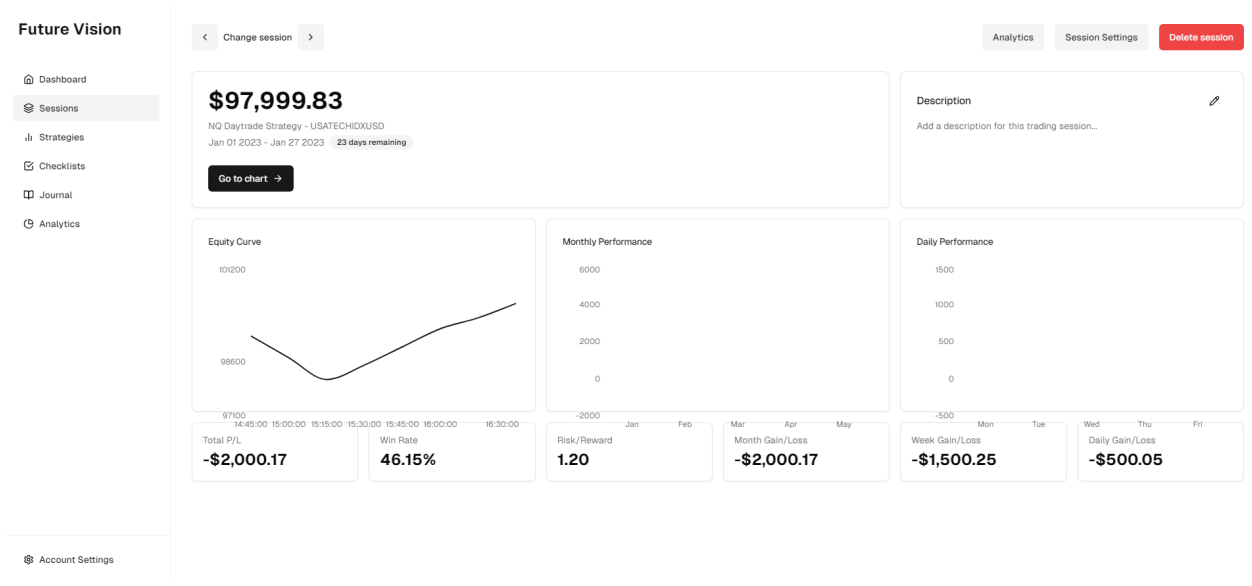

Figure 3: *Session Page of Future Vision*

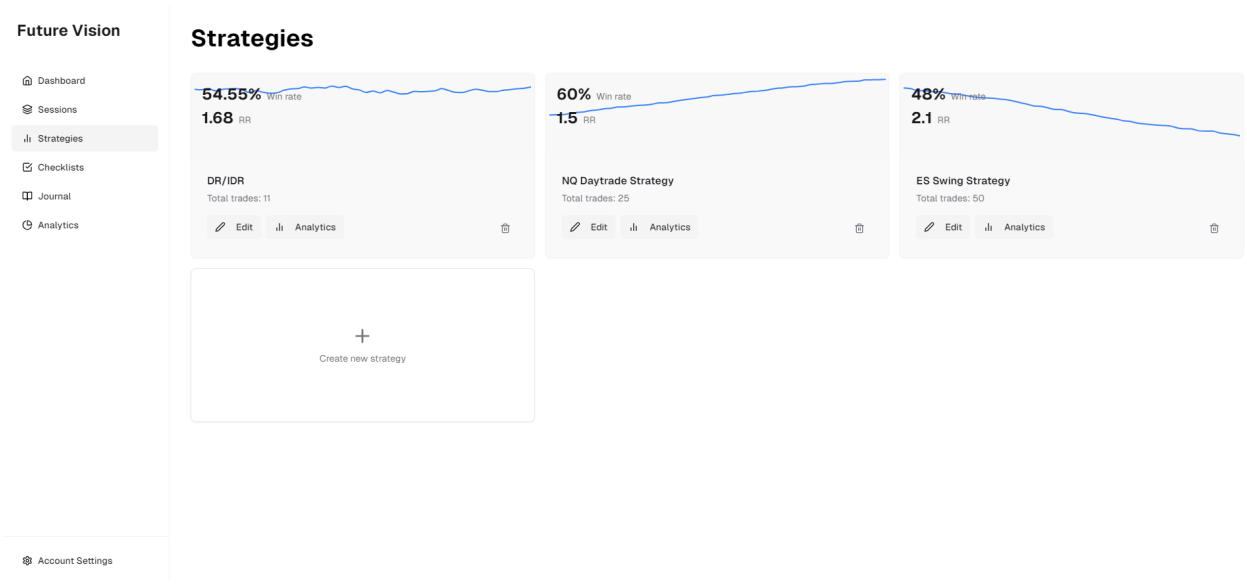

Figure 4: *Strategies Page of Future Vision*

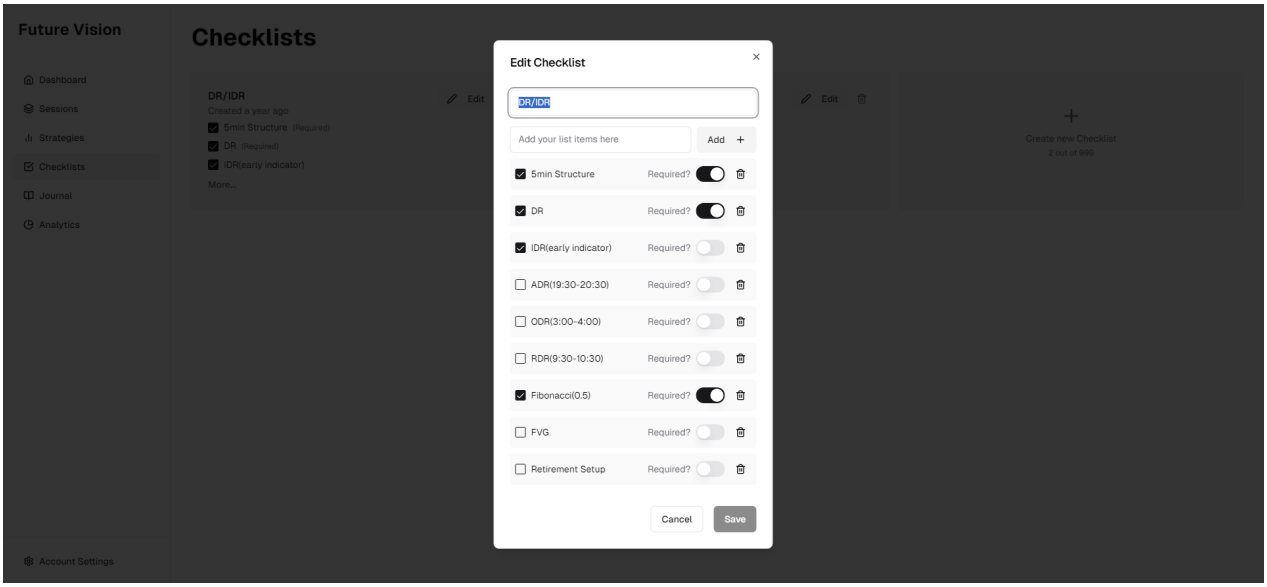

Figure 5: *Checklist of Future Vision*

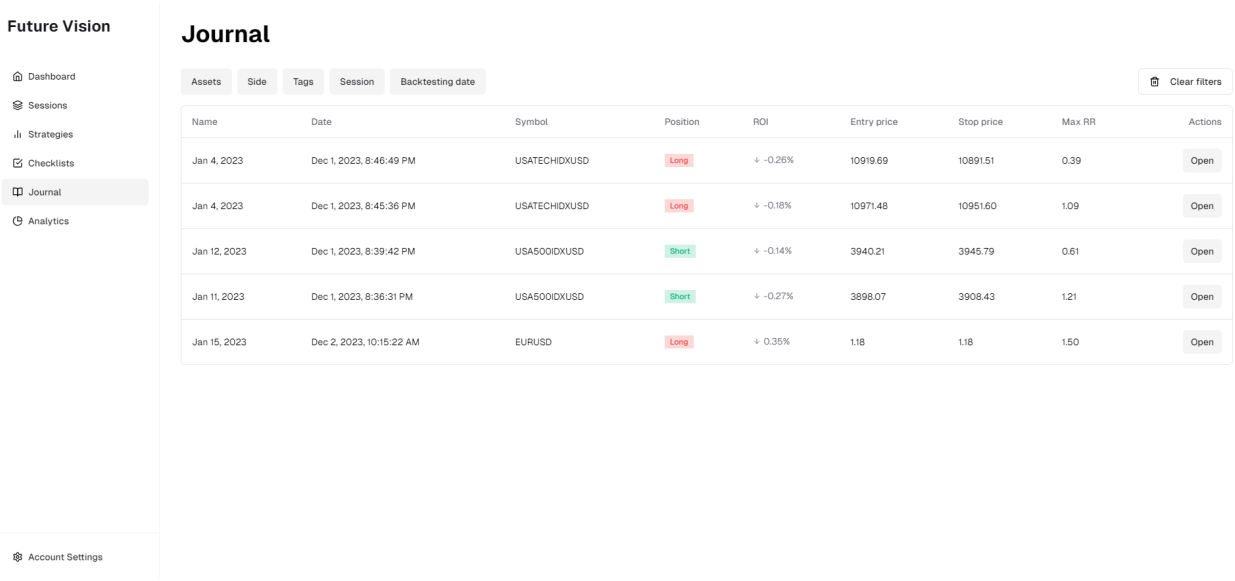

Figure 6: *Journal Page of Future Vision*