**Sprint-4**

**Introduction**

In this Sprint, the purpose was to develop unit tests and integration tests along with backtesting code inorder to test the TRIX Crew AI framework. The following sections contain the User Stories I worked on with a detailed description of the Tasks I worked on.

**User Stories**

I worked on the following User Stories:

[TRIX 1: As a trader, I want to utilize the Trix Indicator integrated with CrewAI agents to identify momentum changes and optimize my trading decisions, so that I can enhance my trading performance and achieve better returns.](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276)

**Conditions of Satisfiability**

**Data Fetching**  
Condition: The system must reliably retrieve accurate real-time and historical stock data.  
Test: Verify data integrity by comparing fetched data against trusted data sources.  
Satisfaction: Data fetched matches external benchmarks with no significant discrepancies.

**Indicator Calculation**  
Condition: Accurate computation of the Trix Indicator using triple-smoothed EMAs and rate of change.  
Test: Compare calculated Trix Indicator values against benchmark calculations for a selected set of stocks.  
Satisfaction: Calculations closely match benchmarks with minimal variance.

**CrewAI Investment Decisions**  
Condition: CrewAI agents must provide relevant and actionable recommendations based on Trix Indicator analysis.  
Test: Input various Trix Indicator scenarios and verify that CrewAI agents generate appropriate buy, sell, or hold recommendations.  
Satisfaction: Recommendations are consistent with Trix Indicator values and align with historical market momentum trends.

**Customization**  
Condition: Users can modify parameters such as smoothing periods.  
Test: Adjust parameters and ensure Trix Indicator calculations and CrewAI recommendations update accordingly.  
Satisfaction: Changes in parameters accurately reflect in both Trix Indicator outputs and investment recommendations without errors.

**Alerts and Notifications**  
Condition: Users can set and receive alerts based on specific Trix Indicator threshold levels.  
Test: Configure alerts for certain Trix Indicator values and verify timely and accurate notifications.  
Satisfaction: Alerts are triggered correctly and delivered promptly without false positives.

**Historical Analysis**  
Condition: The system must accurately analyze and present historical Trix Indicator data to validate current indicators.  
Test: Apply Trix Indicator to historical data and ensure momentum trends are correctly identified and displayed.  
Satisfaction: Historical momentum trends are accurately represented, aiding in the validation of current Trix Indicator signals.

**User Interface**  
Condition: The interface should be user-friendly and facilitate easy interaction with Trix Indicator and CrewAI recommendations.  
Test: Conduct usability testing with traders to ensure the interface is intuitive and meets their needs.  
Satisfaction: Users can efficiently navigate the interface, apply the Trix Indicator, and interpret CrewAI recommendations without difficulty.

**Integration with Trading Platform**  
Condition: Seamless display of CrewAI investment recommendations within existing trading platforms.  
Test: Verify that recommendations are accurately and clearly presented within the trading platform's interface.  
Satisfaction: Integration is smooth, and recommendations are displayed without technical issues or delays.

**Definition of Done**

**Functional Requirements**

* Data Fetching: Implement reliable mechanisms to fetch real-time and historical stock data.
* Calculation Engine: Accurately calculate Trix Indicator values based on fetched data.
* CrewAI Integration: Integrate CrewAI agents to analyze Trix Indicator values and generate investment recommendations.
* Customization Options: Provide user controls to adjust Trix Indicator parameters.
* Alerts System: Develop a system for setting and receiving alerts based on Trix Indicator thresholds.
* Historical Data Analysis: Enable historical analysis of Trix Indicator to validate current momentum trends.
* User Interface: Develop an intuitive interface for applying Trix Indicator and viewing CrewAI recommendations.
* Platform Integration: Ensure seamless integration with existing trading platforms to display recommendations.

**Non-Functional Requirements**

* Performance: The system must process data and update recommendations in real-time with minimal latency.
* Scalability: Capable of handling multiple users and large datasets simultaneously.
* Security: Ensure all data transmissions are secure and comply with industry standards.
* Reliability: System operates consistently without crashes or significant bugs.
* Usability: Interface is user-friendly, reducing the learning curve for new users.
* Compatibility: Functions correctly across various devices and screen sizes, including desktops, tablets, and smartphones.

**Tasks**

* TRIX.1: Implement Indicator Calculation (20 ph)
* TRIX.2: Develop Customization Features (14 ph)
* TRIX.3: Integrate Real-Time and Historical Data (18 ph)
* TRIX.4: Develop Alerts and Notifications System (12 ph)
* TRIX.5: Implement Historical Analysis Capabilities (14 ph)
* TRIX.6: Design and Develop User Interface (20 ph)
* TRIX.7: Integrate Trix Indicator with Trading Platforms (18 ph)
* TRIX.8: Develop Investment Decision Support with CrewAI (16 ph)
* TRIX.9: Ensure Security and Compliance (12 ph)
* TRIX.10: Ensure Performance and Scalability (10 ph)
* TRIX.11: Implement Backtesting Framework (18 ph)
* TRIX.12: Develop Metrics for Backtesting Evaluation (12 ph)
* TRIX.13: Automate Historical Data Selection for Backtesting (10 ph)
* TRIX.14: Implement Forward Testing Framework (16 ph)
* TRIX.15: Develop Real-Time Performance Monitoring for Forward Testing (12 ph)
* TRIX.16: Evaluate and Optimize TRIX-Based Strategies from Forward Testing (14 ph)

**Tasks I Worked On**

[TRIX.10: Ensure Performance and Scalability (10 ph) #370](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/370)

Worked on developing unit tests and integration tests. The task was estimated at 10 person hours but it took me 12 hours to complete.

[TRIX.11: Implement Backtesting Framework (18 ph) #574](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/574)

Worked on backtesting the TRIX Crew AI code. The task was estimated at 18 person hours but it took me 30 hours to complete.

**Summary Table of Work**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UserStory GitHub Issue ID | User Story | Story Points | Task GitHub Issue ID | Task | Task Hours | Status | Actual Hours |
| [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | As a trader, I want to utilize the Trix Indicator integrated with CrewAI agents to identify momentum changes and optimize my trading decisions, so that I can enhance my trading performance and achieve better returns |  | [TRIX.10](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/370) | Ensure Performance and Scalability | 10 | Completed | 12 |
| [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | As a trader, I want to utilize the Trix Indicator integrated with CrewAI agents to identify momentum changes and optimize my trading decisions, so that I can enhance my trading performance and achieve better returns |  | [[TRIX.11](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/574)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/370) | Implement Backtesting Framework | 18 | Completed | 30 |

**Summary Table of Commits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Commit Number | Commit Description (exactly as in github) | User Story | Task |
| April 2nd, 2025 | a80d42908a9de38dd1de597e28037aae862a9123 | [Trix Indicator unit and integration tests](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/556/commits/a80d42908a9de38dd1de597e28037aae862a9123) | [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | [TRIX.10](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/370) |
| April 22nd, 2025 | 642f84063818d5590016433eb23bfb3b59816b60 | [Backtesting Trix Indicator](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/575/commits/642f84063818d5590016433eb23bfb3b59816b60) | [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | [TRIX.11](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/574) |
| April 27th, 2025 | 5a6305420ac46333c220296216b1e19fdf837edf | [Update TRIX backtest code](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/587/commits/5a6305420ac46333c220296216b1e19fdf837edf) | [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | [TRIX.11](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/574) |
| April 27th, 2025 | 91bcd9935f0f050fab9f37747ab8fe714e69e9c5 | [Updated backtest trix code](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/587/commits/91bcd9935f0f050fab9f37747ab8fe714e69e9c5) | [TRIX](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/276) | [TRIX.11](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/574) |