**Sprint-1 Minimal Viable Product**

**Introduction**

In this Sprint, the purpose was to create a minimal viable working product. The following sections contain the User Stories I worked on with a detailed description of the Tasks I worked on.

**User Stories**

[ERI: As a trader, I want to utilize the Elder-Ray Index integrated with CrewAI agents to assess buying and selling pressure and optimize my trading decisions, so that I can enhance my trading performance and achieve better returns. #297](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/297)

**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 Elder-Ray Index using price changes and volume.  
Test: Compare calculated Bull Power and Bear Power 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 Elder-Ray analysis.  
Test: Input various Elder-Ray scenarios and verify that CrewAI agents generate appropriate buy, sell, or hold recommendations.  
Satisfaction: Recommendations are consistent with Elder-Ray values and align with historical market pressure trends.

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

**Alerts and Notifications**  
Condition: Users can set and receive alerts based on specific Elder-Ray threshold levels.  
Test: Configure alerts for certain Elder-Ray 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 Elder-Ray data to validate current indicators.  
Test: Apply Elder-Ray to historical data and ensure buying and selling pressure trends are correctly identified and displayed.  
Satisfaction: Historical pressure trends are accurately represented, aiding in the validation of current Elder-Ray signals.

**User Interface**  
Condition: The interface should be user-friendly and facilitate easy interaction with Elder-Ray 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 Elder-Ray Index, 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 Elder-Ray Index values (Bull Power and Bear Power) based on fetched data.  
CrewAI Integration: Integrate CrewAI agents to analyze Elder-Ray values and generate investment recommendations.  
Customization Options: Provide user controls to adjust Elder-Ray calculation periods.  
Alerts System: Develop a system for setting and receiving alerts based on Elder-Ray thresholds.  
Historical Data Analysis: Enable historical analysis of Elder-Ray to validate current pressure trends.  
User Interface: Develop an intuitive interface for applying Elder-Ray 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.

**Testing and Validation**  
Unit Testing: Test individual components (data fetching, calculation, CrewAI integration) for functionality.  
Integration Testing: Ensure all components work seamlessly together within the trading platform.  
Performance Testing: Evaluate system responsiveness and performance under different loads.  
User Acceptance Testing (UAT): Gather feedback from traders to ensure the system meets their needs and expectations.  
Security Testing: Conduct security audits to verify data protection measures and regulatory compliance.

**Security**  
Data Protection: Encrypt all data transmissions and securely store sensitive information.  
Access Control: Implement role-based access to restrict unauthorized usage of the system.  
Compliance: Adhere to relevant data privacy regulations (e.g., GDPR, CCPA).

**User Experience**  
Intuitive Design: Design the interface for ease of use with clear navigation and controls.  
Feedback Mechanism: Allow users to provide feedback on system functionalities and performance.  
Visual Clarity: Ensure analytical outputs and recommendations are clear and easy to interpret.

**Deployment**  
Staging Environment: Deploy the system to a staging environment for final testing and validation.  
Production Deployment: Launch the system to the production environment without disrupting existing services.  
Monitoring: Continuously monitor system performance and stability post-deployment.  
Bug Fixes: Promptly address and resolve any issues identified during deployment.

**Tasks**

1. [ElderRay.1: Implement Indicator Calculation (20 ph)](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/299)
2. ElderRay.2: Develop Customization Features (14 ph)
3. ElderRay.3: Integrate Real-Time and Historical Data (18 ph)
4. ElderRay.4: Develop Alerts and Notifications System (12 ph)
5. ElderRay.5: Implement Historical Analysis Capabilities (14 ph)
6. ElderRay.6: Design and Develop User Interface (20 ph)
7. ElderRay.7: Integrate Elder-Ray with Trading Platforms (18 ph)
8. ElderRay.8: Develop Investment Decision Support with CrewAI (16 ph)
9. ElderRay.9: Ensure Security and Compliance (12 ph)
10. ElderRay.10: Ensure Performance and Scalability (10 ph)
11. ElderRay.11: Implement Backtesting Framework (18 ph)
12. ElderRay.12: Develop Metrics for Backtesting Evaluation (12 ph)
13. ElderRay.13: Automate Historical Data Selection for Backtesting (10 ph)
14. ElderRay.14: Implement Forward Testing Framework (16 ph)
15. ElderRay.15: Develop Real-Time Performance Monitoring for Forward Testing (12 ph)
16. ElderRay.16: Evaluate and Optimize Elder-Ray-Based Strategies from Forward Testing (14 ph)

**Tasks I Worked On**

[**ElderRay.1: Implement Indicator Calculation (20 ph)**](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/299)

The first task involves developing code for Elder Ray Indicator with Streamlit for UI and the task was estimated at 20 hours but it took me 28 hours.

**Summary Table of Work**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UserStory GitHub Issue ID | User Story | Story Points | Task GitHub Issue ID | Task | Task Hours | Status | Actual Hours |
| [ERI](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/297) | As a trader, I want to utilize the Elder-Ray Index integrated with CrewAI agents to assess buying and selling pressure and optimize my trading decisions, so that I can enhance my trading performance and achieve better returns. |  | [ElderRay.1](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/299) | Implement Indicator Calculation | 20 | Completed | 28 |

**Summary Table of Commits**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Commit Number | Commit Description (exactly as in github) | User Story | Task |
| February 3rd, 2025 | 15b4709cc2b0bca359e8865cf962e7d0afc5c736 | [Add elder ray index file](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/pull/335/commits/15b4709cc2b0bca359e8865cf962e7d0afc5c736) | [ERI](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/297) | [ElderRay.1](https://github.com/Rivier-Computer-Science/AI-Agent-Stock-Prediction/issues/299) |