**Objective:** The goal of this test task is to assess your proficiency in Node.js, API integration, and webhook handling for the development of a Stock Market Signal and Execution Platform. The platform will connect to various brokers, utilize their APIs, and process trading signals from TradingView through webhooks. Additionally, you will need to implement user signup functionality, allowing users to integrate their broker accounts with our platform, specifying ticker/stock preferences, and defining risk and money management rules.

**Task Components:**

1. **Broker API Integration:**
   * Choose a popular broker with a publicly available API (e.g., Alpaca, Interactive Brokers, etc.).
   * Implement a Node.js module to connect to the chosen broker's API.
   * Retrieve account information, including balance and positions.
2. **User Signup and Authentication:**
   * Create a user authentication system with Node.js (you can use frameworks like Express.js).
   * Implement user signup functionality with the following fields:
     + Username
     + Email
     + Password
     + Broker API details (API key, secret, etc.)
3. **Ticker/Stock Preferences:**
   * Allow users to input their preferred tickers or stocks through a simple web interface.
   * Store these preferences securely in the database associated with the user account.
4. **Webhook Handling for TradingView Signals:**
   * Set up an API endpoint to receive webhooks from TradingView.
   * Parse incoming webhook data and extract relevant trading signals (buy/sell/hold recommendations).
5. **Signal Processing and Execution:**
   * Implement logic to process the received trading signals.
   * Use the integrated broker API to execute buy/sell orders based on the signals.
   * Implement basic risk management and money management rules (e.g., maximum position size, stop-loss levels).

**Requirements:**

* Use Node.js and any relevant frameworks (e.g., Express.js).
* Utilize version control (preferably Git) and provide a repository with clear commit history.
* Include clear instructions on how to set up and run your application.
* Prioritize clean and modular code structure.
* Include error handling and logging mechanisms.
* Ensure that sensitive information (API keys, secrets) is securely handled.

**Submission:**

* Provide a link to your Git repository containing the code.
* Include a brief README.md explaining your implementation choices and any additional notes.
* Submit your task within the agreed-upon timeframe.

**Note:** This task is designed to assess your ability to integrate with external APIs, handle webhooks, and implement essential functionalities for a stock market platform. Feel free to make assumptions if specific details are not provided, and document them in your submission. Good luck!