Parlo Dashboard and Web App Requirements for Engineers

General Metrics Display

- **Time Filters for Metrics**:
    - Enable users to filter metrics by various timeframes, allowing them to view data for specific periods such as:
        - **Day**: Display data for the current or selected day.
        - **Month**: Display data for the current or selected month.
        - **Last 3 months**: Aggregate and display data for the last three months.
        - **Last 5 months**: Aggregate and display data for the last five months.
        - **Yearly**: Aggregate and display data for the past year.

Key Metrics to Display

- **Open Active Trades**:
    - Show all trades that are currently open and active in the market.
- **Total Trades Taken**:
    - Display the cumulative number of trades executed by the user or bot.
- **Win/Loss Statistics**:
    - **Number and percentage of winning trades**: Show the count and percentage of trades that were profitable.
    - **Number and percentage of losing trades**: Show the count and percentage of trades that resulted in a loss.
    - **Winning totals in dollars**: Display the total dollar amount gained from winning trades.
- **Current Balance**:
    - Display the user's current account balance.
    - Indicate whether the balance is up or down compared to a previous period, including the percentage change, similar to the display used by Robinhood.

Performance Visualization

- **Performance Graphs**:
    - Graphical representation showing performance metrics by bot, including:
        - **Number of winning trades**: Visualize the count of successful trades.
        - **Revenue generated**: Visualize the total revenue generated from trades.

Trading Log

- **Trading Register**:
    - Chronologically list all trades that have been opened and closed, providing a comprehensive trading history.

Trade Controls

- **Trade Timing**:
    - Provide an option to stop trading a certain time before market close (e.g., 15 minutes before the market closes).
- **Time Control for Bots**:
    - Allow users to set specific trading hours during which the bot is allowed to operate.
    - Configure a mandatory waiting period between trades to avoid rapid successive trades.

Notifications and Alerts

- **Real-time Trade Alerts**:
    - Notify users via text message (SMS) every time a trade is opened or closed. The details should include:
        - **Symbol**: The symbol of the asset being traded.
        - **Entry price**: The price at which the trade was initiated.
        - **Number of contracts**: The number of contracts involved in the trade.
        - **Closing price**: The price at which the trade was closed.
        - **Profit/Loss**: The profit or loss from the trade.
        - **Total P/L for the day**: The cumulative profit or loss for the day.
    - Integration with SMS services (e.g., Twilio) to send these text alerts.
- **Daily Summary**:
    - Send a summary of trading performance at 8 PM EST, including:
        - **Number of trades taken**: Total trades executed during the day.
        - **Number of wins**: Count of profitable trades.
        - **Number of losses**: Count of losing trades.
        - **Profit/Loss**: Total profit or loss for the day.
        - **Commissions**: Total commissions incurred.

Admin Portal

- **Bot Management**:
    - Allow admins to add and manage multiple bots within the platform.
    - Provide options to select the brokerage and type of trading (e.g., futures vs. stock options).
    - Display configuration options based on the selected trading type.
- **Bot Controls**:
    - Set the operational time range for each bot.
    - Define the waiting period between trades to prevent rapid-fire trading.

Additional Features

- **Customizable Dashboard**:
    - Enable users to customize the layout and widgets on their dashboard for a personalized experience.

- **Secure User Management**:
    - Admin capabilities to set up and manage user accounts securely.
    - Implement Two-Factor Authentication (2FA) for enhanced security.
- **Real-time Data and Analysis**:
    - Display real-time market data, including price feeds, order book data, and market depth.
- **Advanced Alerts**:
    - Customizable alerts for significant market movements, order executions, and margin calls.

User Experience Enhancements

- **User-Friendly Interface**:
    - Ensure the platform has an intuitive design for both desktop and mobile web versions.
    - Provide an easy onboarding process with guided tutorials and support.
- **Integrated Communication**:
    - In-platform messaging for support and user community interaction.
- **Educational Resources**:
    - Include a section for tutorials, guides, and FAQs to help users understand and effectively use the platform.

Technical Implementation Notes:

1. **API Integration**:
    - Secure API integrations with TradingView for receiving webhooks.
    - Integrate with Tradovate for executing trades.
    - Integrate with SMS services (e.g., Twilio) for sending text alerts.
2. **Backend Development**:
    - Develop a robust backend using a scalable technology stack (e.g., Node.js, Python, AWS).
    - Implement real-time data processing and storage solutions.
3. **Frontend Development**:
    - Use modern frontend frameworks (e.g., React, Angular) to create a responsive and intuitive UI.
    - Ensure seamless user experience across both desktop and mobile web versions.
4. **Security**:
    - Implement comprehensive security measures including encryption, secure APIs, and 2FA.
    - Ensure compliance with industry standards for data protection and security.
5. **Performance Optimization**:
    - Optimize data fetching and rendering to ensure real-time performance.
    - Implement caching strategies and load balancing to handle high traffic.