

# Product Requirements Document

---

**Jota Crowd**  
Crowdbotics, Inc.

# Table of contents

Feature	Page
<b>Application</b>	<b>1.0</b>
Description	1.1
Goals & Objectives	1.2
User Types	1.3
<b>Onboarding &amp; User Management</b>	<b>2</b>
Sign-up process for traders and investors	2.1
Profile customization for user preferences	2.2
Role-based access control for different user types (Active Trader, Passive Investor, Fund Manager)	2.3
Multi-factor authentication for enhanced security	2.4
Interactive onboarding tutorial for platform navigation	2.5
Privacy settings introduction	2.6
Bulk user import for fund managers managing multiple accounts	2.7
User activity and audit logs for compliance and monitoring	2.8
Automated onboarding workflows for new users	2.9
Accessibility settings for users with disabilities	2.10
Customizable user dashboards based on trading preferences	2.11
<b>AI Agent Architecture &amp; Functionality</b>	<b>3</b>
Indicator Analysis Agents for monitoring technical indicators (e.g., RSI, MACD, Bollinger Bands)	3.1
Strategy Evaluation Agents for assessing trading strategies (e.g., scalping, swing trading)	3.2
Probability Assessment Agents for calculating trade success likelihood	3.3
Execution Agents for automated trade placement and management	3.4
Inter-agent communication framework with standardized data exchange	3.5
Feedback mechanism for refining agent decision-making based on trade outcomes	3.6
Dynamic agent scaling to handle high market data volumes	3.7
Real-time volatility monitoring and adjustment by agents	3.8
AI-driven insights for portfolio optimization	3.9
Cross-agent collaboration for complex trading scenarios	3.10
<b>Exchange Integration</b>	<b>4</b>
Standardized API integration framework for market data retrieval and order placement	4.1
Exchange-specific adapters for unique features and limitations	4.2
Integration with major exchanges (Binance, Coinbase Pro, Kraken, FTX)	4.3
Roadmap for additional exchange integrations (10+ exchanges within 12 months)	4.4

Rate limiting compliance for API calls	4.5
Error handling for exchange-specific issues	4.6
Real-time account balance monitoring	4.7
Support for multiple trading pairs across exchanges	4.8
On-chain transaction data integration for fundamental analysis	4.9
Cross-exchange arbitrage capabilities	4.10

## Trading Strategies

**5**

Implementation of core trading strategies (e.g., scalping, swing trading, trend following)	5.1
Strategy customization interface with adjustable parameters	5.2
Real-time and historical performance analytics for strategies	5.3
Backtesting engine using historical data across multiple timeframes	5.4
Simulation mode for paper trading with real-time market data	5.5
Dynamic position sizing based on market volatility	5.6
Portfolio optimization tools for risk-adjusted returns	5.7
AI-driven strategy recommendations based on market conditions	5.8
Social trading features for sharing and copying strategies	5.9
Customizable strategy templates for advanced users	5.10

## Risk Management

**6**

Maximum trade size controls (% of portfolio)	6.1
Stop-loss settings (fixed and trailing)	6.2
Take-profit settings	6.3
Maximum daily trading volume limits	6.4
Maximum drawdown thresholds	6.5
Risk visualization dashboard for current exposure	6.6
Risk alert system with notifications via in-app, email, and SMS	6.7
Circuit breaker mechanism for automatic trading pause during extreme volatility	6.8
Volatility-adjusted position sizing	6.9
AI-driven risk mitigation strategies	6.10

## Analytics & Reporting

**7**

Comprehensive dashboards for trading performance insights	7.1
Customizable report builder for trading metrics	7.2
Performance benchmarking against market indices and manual trading	7.3
Predictive analytics for trading success and market trends	7.4
Real-time analytics dashboard for trade execution and performance	7.5

Historical trade analysis with detailed metrics (win rate, profit factor, drawdown)	7.6
Heat maps for market activity and volatility	7.7
Integration with tax reporting tools for financial compliance	7.8
Social impact reporting for ethical trading practices	7.9
AI-driven insights for improving trading strategies	7.10

**User Experience****8**

Intuitive step-by-step onboarding process	8.1
Simplified dashboard for beginner users with recommended settings	8.2
Advanced configuration tools for experienced traders	8.3
Customizable user interface with drag-and-drop widgets	8.4
Dark mode for reduced eye strain	8.5
Voice-activated navigation and commands	8.6
Mobile app synchronization for seamless access	8.7
Push notifications for trade updates and alerts	8.8
Interactive tutorials for platform features	8.9
Accessibility features for users with disabilities	8.10

**Security & Compliance****9**

Secure storage and handling of exchange API keys (AES-256 encryption)	9.1
Multi-factor authentication system (app-based 2FA, email verification, biometric authentication)	9.2
Quarterly third-party security audits with published results	9.3
Data encryption at rest and in transit	9.4
Role-based access control for sensitive features	9.5
Blockchain-based data integrity and audit trails	9.6
Real-time threat detection and response systems	9.7
Compliance with relevant cryptocurrency regulations	9.8
Anomaly detection systems for proactive security measures	9.9
Secure payment processing for subscription and performance fees	9.10

**Mobile App****10**

Cross-platform mobile app for iOS and Android	10.1
Offline access to trading data and strategies	10.2
Push notifications for trade updates and risk alerts	10.3
Mobile-optimized user interface for all features	10.4
Biometric authentication for enhanced security	10.5
Augmented reality features for market visualization	10.6

Voice commands for hands-free operation	10.7
Integration with smartwatches and wearable devices	10.8
Interactive tutorials for mobile app navigation	10.9
Customizable notification preferences for users	10.10

## **Subscription & Pricing** **11**

Multiple subscription tiers with differentiated features and limits	11.1
Performance-based fee structure with high-water mark tracking	11.2
Secure billing system supporting multiple payment methods (including cryptocurrency)	11.3
Free trial period for new users	11.4
Discounts for long-term subscriptions	11.5
Customizable pricing plans for enterprise users	11.6
Automated invoicing and payment reminders	11.7
Integration with financial software for subscription management	11.8
Transparent fee breakdown for users	11.9
AI-driven recommendations for optimal subscription tier	11.10

## **Non-Functional Requirements** **12**

Scalability and Performance	12.1
Cross-Platform Compatibility	12.2
Security and Data Protection	12.3
High Availability and Disaster Recovery	12.4
Latency for AI Agent Operations	12.5
Accessibility Compliance	12.6
Integration Reliability	12.7
Data Analytics Accuracy	12.8
Subscription Management Efficiency	12.9
User Experience Optimization	12.10

# Application

## Description

Not provided

## User Types

- Active Trader
- Passive Investor
- Fund Manager

# Phase 1

## Onboarding & User Management

### Sign-up process for traders and investors

#### Description

The 'Sign-up process for traders and investors' feature in the 'my-test-juan-0304' app provides a streamlined and secure onboarding experience for new users. This feature allows individuals to create accounts tailored to their specific roles (Active Trader, Passive Investor, or Fund Manager) and preferences. The sign-up process is designed to be intuitive, ensuring users can quickly register and access the platform's functionalities. It incorporates multi-factor authentication for enhanced security and collects essential information to personalize the user experience. The feature also supports bulk user import for fund managers managing multiple accounts, making it efficient for professional use cases.

#### Acceptance Criteria

- The sign-up process must be accessible from the app's main landing page and mobile app, with clear call-to-action buttons.
- Users must be able to select their role (Active Trader, Passive Investor, or Fund Manager) during the sign-up process, with tailored onboarding flows for each role.
- The sign-up form must collect essential information, including name, email, password, and role-specific details (e.g., trading preferences for traders, portfolio management details for fund managers).
- Multi-factor authentication must be implemented, supporting app-based 2FA, email verification, and biometric authentication for enhanced security.
- The sign-up process must include validation checks for input fields, ensuring data accuracy and preventing errors (e.g., invalid email formats, weak passwords).
- Fund managers must have the option to bulk import user accounts, with support for CSV file uploads and automated account creation for multiple users.
- The onboarding flow must include an interactive tutorial for platform navigation, tailored to the user's selected role.
- Privacy settings must be introduced during the sign-up process, allowing users to configure their data-sharing preferences and account visibility.
- Accessibility settings must be available during sign-up, adhering to WCAG standards to ensure usability for users with disabilities.
- The sign-up process must demonstrate high performance, with quick response times and minimal delays, even during peak usage periods.
- Error handling must be implemented to manage potential issues, such as network connectivity problems or duplicate account creation attempts, with user-friendly feedback messages.
- The sign-up feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Analytics must be integrated to track user sign-up patterns and behavior, providing insights for improving the onboarding experience.
- The sign-up interface must be visually appealing and intuitive, with a responsive design that adapts to various screen sizes and resolutions.

### Profile customization for user preferences

#### Description

The 'Profile customization for user preferences' feature in the 'my-test-juan-0304' app allows users to personalize their profiles to align with their trading and investment goals. This feature provides a user-friendly interface for setting preferences such as trading strategies, risk tolerance, notification settings, and dashboard layout. By tailoring the app experience to individual needs, this feature enhances usability and ensures that users can efficiently access the tools and information most relevant to their roles and objectives.

#### Acceptance Criteria

- Users must be able to access the profile customization feature from their account settings or dashboard.
- The customization interface should allow users to set preferences for trading strategies, risk tolerance, and notification types (e.g., email, SMS, in-app).
- Users must be able to customize their dashboard layout by adding, removing, or rearranging widgets to prioritize relevant information.
- The app should provide pre-configured templates for different user roles (Active Trader, Passive Investor, Fund Manager) to simplify the customization process.
- Changes made to user preferences must be saved in real-time and persist across sessions.
- The app should provide a preview mode for users to see how their customized dashboard will appear before saving changes.
- Users must be able to reset their preferences to default settings with a single action.
- The customization feature must be accessible and adhere to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

- The app should provide contextual help or tooltips to guide users through the customization process.
- The feature must support multi-device synchronization, ensuring that customizations made on one device are reflected across all devices linked to the user's account.
- The app should include analytics to track user engagement with the customization feature, providing insights for future improvements.

## Role-based access control for different user types (Active Trader, Passive Investor, Fund Manager)

### Description

The 'Role-based access control for different user types' feature in the 'my-test-juan-0304' app ensures that users are granted access to functionalities and resources based on their designated roles: Active Trader, Passive Investor, or Fund Manager. This feature enhances security and usability by tailoring the app experience to the specific needs and permissions of each user type. Active Traders can access advanced trading tools and real-time analytics, Passive Investors are provided with simplified dashboards and portfolio management features, and Fund Managers are equipped with bulk account management and reporting capabilities. This role-based system ensures compliance, prevents unauthorized access, and optimizes the user experience for each role.

### Acceptance Criteria

- The app must allow users to select their role (Active Trader, Passive Investor, Fund Manager) during the onboarding process.
- Each user role must have distinct access permissions and features tailored to their needs, ensuring a personalized experience.
- Active Traders must have access to advanced trading tools, real-time analytics, and strategy customization interfaces.
- Passive Investors must have access to simplified dashboards, portfolio management tools, and AI-driven recommendations for investment strategies.
- Fund Managers must have access to bulk account management features, customizable reporting tools, and compliance monitoring dashboards.
- The app must prevent users from accessing features or data outside their designated role permissions, ensuring security and compliance.
- Role-based access control must be implemented at both the front-end and back-end levels to ensure robust security.
- Administrators must have the ability to modify user roles and permissions in case of role changes or errors.
- The app must provide clear documentation and tutorials for each role, explaining the features and permissions available to them.
- Role-based access control must be tested across different devices and operating systems to ensure consistent functionality.
- The app must log user activity and access attempts for compliance and monitoring purposes, ensuring accountability.
- Error handling must be implemented to provide feedback to users attempting to access unauthorized features, with friendly messages explaining the restrictions.
- The role-based access control system must be scalable to accommodate additional roles or permissions in the future.

## Multi-factor authentication for enhanced security

### Description

The 'Multi-factor authentication for enhanced security' feature in the 'my-test-juan-0304' app provides an additional layer of protection for user accounts by requiring multiple forms of verification during login. This feature ensures that only authorized users can access sensitive trading and account information, significantly reducing the risk of unauthorized access. It supports various authentication methods, including app-based 2FA, email verification, and biometric authentication, offering flexibility and convenience for users. By implementing multi-factor authentication, the app enhances overall security and builds user trust, especially in the high-stakes environment of trading and investment.

### Acceptance Criteria

- Users must be able to enable multi-factor authentication (MFA) from their account settings, with clear instructions provided for setup.
- The MFA system must support app-based 2FA (e.g., Google Authenticator, Authy), email verification, and biometric authentication (e.g., fingerprint or facial recognition).
- During login, users must be prompted to complete the second authentication step after entering their password.
- The app must provide fallback options for MFA, such as backup codes or alternative verification methods, in case the primary method is unavailable.
- Users must be notified via email or in-app notifications when MFA is enabled, disabled, or when there are changes to their authentication settings.
- The MFA system must demonstrate high reliability and quick response times, ensuring a seamless user experience without significant delays.
- Error handling must be implemented to manage issues such as incorrect codes, expired codes, or failed biometric scans, with clear feedback provided to users.
- The app must ensure that MFA settings are securely stored and encrypted, protecting user data from potential breaches.
- MFA must be mandatory for certain high-risk actions, such as withdrawing funds, changing account settings, or linking exchange APIs.
- The MFA system must comply with relevant security standards and regulations, ensuring robust protection against unauthorized access.



- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The MFA interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Interactive onboarding tutorial for platform navigation

### Description

The 'Interactive onboarding tutorial for platform navigation' feature in the 'my-test-juan-0304' app is designed to provide new users with a seamless and engaging introduction to the platform. This tutorial guides users through the core functionalities and features of the app, ensuring they understand how to navigate and utilize the tools available for trading, investing, and portfolio management. The tutorial is interactive, allowing users to explore the platform while receiving step-by-step instructions, tips, and contextual guidance tailored to their specific user role. By simplifying the onboarding process, this feature enhances user confidence and accelerates their ability to effectively use the app.

### Acceptance Criteria

- The onboarding tutorial must be automatically triggered for new users upon their first login, with an option to skip or revisit it later.
- The tutorial must adapt its content based on the user's role (Active Trader, Passive Investor, Fund Manager), highlighting relevant features and workflows.
- Interactive elements, such as clickable hotspots, tooltips, and guided walkthroughs, must be included to provide hands-on learning experiences.
- The tutorial must cover key platform functionalities, including dashboard navigation, trading tools, analytics, risk management settings, and subscription management.
- Users must be able to pause, resume, or restart the tutorial at any time from the settings menu.
- The tutorial must include visual aids, such as animations, diagrams, and videos, to enhance understanding and engagement.
- Progress tracking must be implemented, allowing users to see which sections of the tutorial they have completed and what remains.
- The tutorial must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for users with disabilities.
- Feedback mechanisms, such as surveys or ratings, must be included at the end of the tutorial to gather user input for future improvements.
- The tutorial must demonstrate high performance, loading quickly and functioning smoothly across different devices and operating systems.
- Error handling must be implemented to ensure the tutorial remains functional in cases of network connectivity issues or app crashes.
- Analytics must be integrated to track user engagement with the tutorial, providing insights into completion rates and areas of difficulty.
- The tutorial must be compatible with the latest version of the app and tested across various screen sizes and resolutions.

## Privacy settings introduction

### Description

The 'Privacy settings introduction' feature in the 'my-test-juan-0304' app provides users with a clear and concise overview of the privacy settings available within the platform. During onboarding, users are guided through the various options to control their data visibility, sharing preferences, and account security. This feature ensures that users understand how their data is handled and empowers them to customize their privacy settings according to their preferences. It is designed to enhance user trust and compliance with data protection regulations.

### Acceptance Criteria

- The privacy settings introduction must be presented as part of the onboarding process for all new users, ensuring it is easily accessible and not skipped.
- Users must be able to view a summary of the app's data handling policies, including how their data is stored, shared, and used.
- The feature must include an interactive walkthrough or tutorial that explains key privacy settings, such as profile visibility, data sharing preferences, and notification settings.
- Users must be able to customize their privacy settings directly from the introduction screen, with options to adjust visibility (e.g., public, private, or restricted), data sharing permissions, and communication preferences.
- The app must provide clear explanations of each privacy setting, including the implications of enabling or disabling specific options, ensuring users make informed decisions.
- The privacy settings introduction must comply with relevant data protection regulations, such as GDPR, CCPA, or other applicable laws, and include links to the app's privacy policy for further details.
- Users must be able to revisit the privacy settings introduction at any time from the settings menu, allowing them to update their preferences as needed.
- The feature must include a confirmation step where users acknowledge their understanding of the privacy settings and agree to the app's data handling policies.
- The privacy settings introduction must be compatible with all user roles (Active Trader, Passive Investor, Fund Manager), ensuring tailored explanations and options based on the role-specific data usage.

- The interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.
- The app must provide feedback to users when privacy settings are successfully updated, ensuring transparency and user confidence.
- Include error handling to manage potential issues, such as network connectivity problems or server errors, ensuring the feature remains stable and provides appropriate feedback to users.

## Bulk user import for fund managers managing multiple accounts

### Description

The 'Bulk user import for fund managers managing multiple accounts' feature in the 'my-test-juan-0304' app streamlines the onboarding process for fund managers by enabling them to efficiently import multiple user accounts in a single operation. This feature is designed to save time and reduce manual effort by allowing fund managers to upload user data in bulk using standardized file formats (e.g., CSV, Excel). The system validates the uploaded data, ensuring accuracy and compliance with platform requirements. Once imported, fund managers can easily manage these accounts, assign roles, and customize settings, enhancing operational efficiency and scalability.

### Acceptance Criteria

- Fund managers must be able to access the bulk user import functionality from their dashboard or user management section.
- The feature must support standardized file formats for bulk uploads, including CSV and Excel, with clear templates provided for formatting user data.
- Uploaded files must undergo validation to check for errors, such as missing fields, incorrect data formats, or duplicate entries, with detailed feedback provided to the fund manager for corrections.
- The system must ensure compliance with platform requirements, such as role-based access control and data security standards, during the import process.
- Fund managers must be able to assign roles (e.g., Active Trader, Passive Investor) and customize account settings for imported users during or after the import process.
- The feature must provide a summary of the import process, including the number of successfully imported accounts, errors encountered, and actions taken.
- Imported user accounts must be immediately accessible within the platform, allowing fund managers to manage them seamlessly.
- The system must include error handling for scenarios such as file upload failures, network issues, or data validation errors, with appropriate feedback provided to the user.
- The bulk user import process must demonstrate high performance, handling large datasets efficiently without significant delays.
- The feature must adhere to data privacy and security standards, ensuring that sensitive user information is encrypted and protected during the import process.
- Interactive tutorials or help documentation must be available to guide fund managers through the bulk user import process.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.

## User activity and audit logs for compliance and monitoring

### Description

The 'User activity and audit logs for compliance and monitoring' feature in the 'my-test-juan-0304' app provides a comprehensive logging system to track and record user actions within the platform. This feature is designed to ensure compliance with regulatory requirements, enhance transparency, and provide insights into user behavior. It enables administrators and fund managers to monitor activities such as trade execution, account changes, and system interactions. The logs are securely stored and can be filtered by user role, date, and activity type, ensuring ease of access and analysis. This feature is essential for maintaining accountability, detecting anomalies, and supporting audits.

### Acceptance Criteria

- The system must log all user activities, including but not limited to trade execution, account modifications, login/logout events, and changes to settings.
- Logs must include detailed information such as timestamp, user ID, activity type, and relevant metadata (e.g., trade details, IP address).
- The logs must be accessible to authorized users, such as fund managers and administrators, through a secure interface with role-based access control.
- The system must provide filtering and search capabilities to allow users to easily locate specific logs based on criteria such as date range, user role, or activity type.
- Logs must be stored securely using encryption at rest and in transit, ensuring data integrity and protection against unauthorized access.
- The system must comply with relevant regulatory standards for data retention and audit logging, including GDPR, SEC, and other applicable guidelines.
- The logs must include an anomaly detection mechanism to flag suspicious activities, such as unusual trading patterns or unauthorized access attempts.
- The system must provide export functionality for logs in standard formats (e.g., CSV, JSON) to support external audits and compliance reporting.

- The logs must be retained for a configurable period, with options for automatic archival or deletion based on organizational policies.
- The system must include error handling to ensure logging continues uninterrupted during network or system failures, with fallback mechanisms for data recovery.
- The interface for viewing logs must be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards.
- The system must demonstrate high performance, allowing for real-time logging and retrieval of large datasets without significant delays.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.

## Automated onboarding workflows for new users

### Description

The 'Automated onboarding workflows for new users' feature in the 'my-test-juan-0304' app streamlines the onboarding process for new users by automating key steps based on their selected user role (Active Trader, Passive Investor, or Fund Manager). This feature ensures a personalized and efficient onboarding experience by dynamically tailoring the workflow to match the user's preferences, trading goals, and role-specific requirements. It includes automated account setup, role-based tutorials, and pre-configured settings to help users quickly familiarize themselves with the platform and start using its features effectively.

### Acceptance Criteria

- The onboarding workflow must automatically identify the user's role (Active Trader, Passive Investor, or Fund Manager) during the sign-up process and customize the onboarding steps accordingly.
- The workflow must include automated account setup, such as linking exchange accounts, configuring default settings, and enabling relevant features based on the user's role.
- Role-specific tutorials must be provided during onboarding, offering interactive guidance tailored to the user's trading goals and platform navigation needs.
- The onboarding process must include pre-configured dashboards and settings optimized for each user role, ensuring a seamless start to using the app.
- Users must be able to skip or revisit specific onboarding steps, providing flexibility in how they complete the process.
- The onboarding workflow must integrate with accessibility settings to ensure usability for users with disabilities, adhering to WCAG standards.
- Automated workflows must demonstrate high performance and reliability, ensuring quick response times and minimal errors during the onboarding process.
- Error handling mechanisms must be implemented to manage potential issues, such as network connectivity problems or incomplete user information, with clear feedback provided to the user.
- The onboarding process must include a progress tracker, allowing users to see their completion status and remaining steps.
- Analytics must be integrated to track user engagement and completion rates during onboarding, providing insights for further optimization of the workflows.
- The feature must be compatible with both desktop and mobile versions of the app, ensuring consistency across devices.
- The onboarding workflows must comply with relevant data privacy and security regulations, ensuring that user information is handled securely.

## Accessibility settings for users with disabilities

### Description

The 'Accessibility settings for users with disabilities' feature in the 'my-test-juan-0304' app ensures an inclusive user experience by providing customizable accessibility options tailored to the needs of users with disabilities. This feature includes tools and settings that enhance usability, such as screen reader compatibility, adjustable text sizes, high-contrast themes, keyboard navigation, and voice commands. By prioritizing accessibility, the app aims to empower all users, regardless of physical or cognitive limitations, to effectively navigate and utilize its trading and investment functionalities.

### Acceptance Criteria

- The app must include a dedicated accessibility settings section within the user profile or settings menu, easily accessible to all users.
- Users must be able to enable screen reader compatibility, ensuring that all app elements are properly labeled and navigable using assistive technologies.
- Provide adjustable text size options, allowing users to increase or decrease font sizes for better readability.
- Include high-contrast themes to improve visibility for users with visual impairments, ensuring all text and UI elements are clearly distinguishable.
- Support keyboard navigation for users who cannot use a mouse or touch interface, with clear focus indicators for active elements.
- Implement voice command functionality for hands-free navigation and operation of key app features.
- Ensure all interactive elements, such as buttons and links, are large enough and spaced appropriately for users with motor impairments.
- Adhere to WCAG (Web Content Accessibility Guidelines) standards to ensure compliance with accessibility best practices.
- Provide an onboarding tutorial specifically for accessibility settings, guiding users on how to customize the app to meet their needs.

- Test the accessibility settings across multiple devices and operating systems to ensure consistent functionality and usability.
- Include error handling and feedback mechanisms to inform users if accessibility features encounter issues or require updates.
- Ensure that enabling accessibility settings does not compromise the app's performance or security features.
- Integrate analytics to monitor the usage of accessibility settings, enabling continuous improvement based on user feedback and behavior.

## Customizable user dashboards based on trading preferences

### Description

The 'Customizable user dashboards based on trading preferences' feature in the 'my-test-juan-0304' app allows users to tailor their dashboard experience to align with their specific trading needs and preferences. This feature provides flexibility for users to select and organize widgets, metrics, and tools that are most relevant to their trading strategies and goals. Active Traders can focus on real-time market data and execution tools, Passive Investors can prioritize portfolio performance and long-term analytics, and Fund Managers can emphasize multi-account management and compliance tracking. The feature enhances user experience by ensuring that each user role has access to a personalized and efficient workspace, improving productivity and decision-making.

### Acceptance Criteria

- Users must be able to access the dashboard customization interface from the main dashboard screen.
- The customization interface should allow users to add, remove, and rearrange widgets using a drag-and-drop functionality.
- Widgets must include options such as real-time market data, portfolio performance metrics, risk management tools, trade execution shortcuts, and compliance tracking features.
- Each user role (Active Trader, Passive Investor, Fund Manager) must have a default dashboard layout tailored to their typical needs, which can be further customized by the user.
- Users must be able to save multiple dashboard layouts and switch between them based on their current trading focus or strategy.
- The dashboard customization interface must include a preview mode, allowing users to see how their changes will look before saving them.
- The app must provide recommendations for widgets based on the user's trading preferences and historical behavior, using AI-driven insights.
- Customizations must be stored securely and synced across devices, ensuring consistency for users accessing the app on multiple platforms (e.g., desktop, mobile).
- The dashboard must support real-time updates for widgets displaying dynamic data, such as market prices and trade execution statuses.
- Accessibility features must be integrated into the dashboard customization interface, adhering to WCAG standards to ensure usability for all users, including those with disabilities.
- The feature must be tested for compatibility across different devices and operating systems to ensure a consistent user experience.
- Error handling must be implemented to manage issues such as failed widget loading or synchronization errors, providing clear feedback to the user.
- Analytics must be integrated to track user interactions with the dashboard customization feature, enabling continuous improvement based on user behavior and feedback.

## AI Agent Architecture & Functionality

### Indicator Analysis Agents for monitoring technical indicators (e.g., RSI, MACD, Bollinger Bands)

#### Description

The 'Indicator Analysis Agents for monitoring technical indicators' feature in the 'my-test-juan-0304' app provides advanced AI-driven agents that continuously monitor and analyze key technical indicators such as RSI (Relative Strength Index), MACD (Moving Average Convergence Divergence), and Bollinger Bands. These agents are designed to assist users in identifying potential trading opportunities, market trends, and entry/exit points by providing real-time insights and actionable recommendations. The feature is tailored for users who rely on technical analysis to make informed trading decisions, ensuring precision and efficiency in their strategies.

#### Acceptance Criteria

- The Indicator Analysis Agents must support monitoring of key technical indicators, including RSI, MACD, and Bollinger Bands, with the ability to expand to additional indicators based on user demand.
- Agents must provide real-time analysis and updates, ensuring users have access to the latest market data and insights.
- The feature must include customizable thresholds and parameters for each indicator, allowing users to tailor the analysis to their specific trading strategies.
- Users must be able to view indicator trends and signals through a visually intuitive dashboard, including charts and graphs for easy interpretation.
- The agents should generate actionable recommendations, such as buy/sell signals, based on the analysis of technical indicators and predefined user preferences.

- Integration with the app's notification system must be implemented, enabling users to receive alerts for significant indicator changes or signals via in-app notifications, email, or SMS.
- The feature must demonstrate high performance and reliability, ensuring accurate calculations and minimal latency in data processing.
- Indicator Analysis Agents must be compatible with multiple trading pairs and exchanges, providing comprehensive coverage for diverse trading scenarios.
- The feature must include error handling mechanisms to address potential issues with data retrieval or processing, ensuring stability and user feedback in case of errors.
- Ensure the agents adhere to user privacy and data security standards, with no sensitive information exposed during analysis or reporting.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and user experience.
- Provide interactive tutorials and documentation to help users understand and utilize the Indicator Analysis Agents effectively.
- Include analytics to track user engagement with the feature and refine the agents' performance based on user feedback and behavior.

## Strategy Evaluation Agents for assessing trading strategies (e.g., scalping, swing trading)

### Description

The 'Strategy Evaluation Agents for assessing trading strategies' feature in the 'my-test-juan-0304' app leverages AI-driven agents to analyze and evaluate the effectiveness of various trading strategies, such as scalping, swing trading, and trend following. These agents utilize historical and real-time market data to provide insights into the performance, risk, and profitability of each strategy. By offering detailed evaluations, users can make informed decisions about which strategies align with their trading goals and risk tolerance. This feature is designed to enhance trading efficiency and optimize portfolio management for all user roles.

### Acceptance Criteria

- The Strategy Evaluation Agents must be accessible from the main dashboard or a dedicated 'AI Agents' section within the app.
- Users must be able to select specific trading strategies (e.g., scalping, swing trading) for evaluation, with options to customize parameters such as timeframes and risk levels.
- The agents should analyze both historical and real-time market data to provide comprehensive evaluations of strategy performance.
- Evaluation results must include key metrics such as win rate, profit factor, drawdown, average trade duration, and risk-adjusted returns.
- The feature should offer visualizations, such as charts and graphs, to help users understand the performance trends and risk profiles of each strategy.
- Users must be able to compare multiple strategies side-by-side to identify the most suitable option for their trading objectives.
- The agents should provide actionable recommendations, such as adjustments to strategy parameters or alternative strategies based on market conditions.
- The feature must support integration with the backtesting engine, allowing users to simulate strategy performance using historical data before live implementation.
- Ensure high performance and quick response times for strategy evaluations, even when processing large datasets or complex strategies.
- Include error handling to manage potential issues with data retrieval or processing, providing clear feedback to users about any errors encountered.
- The feature must adhere to user privacy and data security standards, ensuring sensitive trading data is protected during analysis.
- Provide interactive tutorials or tooltips to guide users in utilizing the Strategy Evaluation Agents effectively.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.
- The interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Probability Assessment Agents for calculating trade success likelihood

### Description

The 'Probability Assessment Agents for calculating trade success likelihood' feature in the 'my-test-juan-0304' app leverages advanced AI algorithms to evaluate the probability of success for individual trades. By analyzing historical data, market conditions, technical indicators, and user-defined parameters, these agents provide actionable insights to help users make informed trading decisions. The feature is designed to enhance decision-making by offering probabilistic assessments that quantify the likelihood of achieving profitable outcomes for specific trades. This functionality is critical for risk management and strategy optimization, enabling users to align their trading actions with data-driven predictions.

### Acceptance Criteria

- The Probability Assessment Agents must utilize historical market data, technical indicators, and real-time market conditions to calculate trade success likelihood.

- Users must be able to input custom parameters, such as risk tolerance, trade duration, and target profit, to refine the probability assessments.
- The agents should provide a clear and intuitive visualization of probability scores, such as percentage likelihood or confidence intervals, for each trade opportunity.
- The feature must support integration with other AI agents, such as Strategy Evaluation Agents and Execution Agents, to provide a seamless trading experience.
- Probability assessments must be updated in real-time to reflect changing market conditions and user inputs, ensuring accuracy and relevance.
- The app should include a detailed explanation of how probability scores are calculated, ensuring transparency and user trust in the AI-driven insights.
- Users must be able to access probability assessments directly from their dashboard or trading interface, ensuring ease of use and accessibility.
- The feature must demonstrate high performance and quick response times, effectively handling large datasets and complex calculations without significant delays.
- Include error handling to manage potential failures in data retrieval or algorithmic processing, ensuring the app remains stable and provides feedback to the user about the error.
- The probability assessment algorithm should prioritize user privacy and data security, ensuring that sensitive information is not exposed or misused.
- Integrate analytics to track and analyze user interactions with the Probability Assessment Agents, which can be used to further refine and improve the feature.
- Ensure the feature is compatible with the latest version of the app and is tested across different devices and operating systems for consistency.
- The probability assessment interface should be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Execution Agents for automated trade placement and management

### Description

The 'Execution Agents for automated trade placement and management' feature in the 'my-test-juan-0304' app empowers users to automate the execution of trades based on predefined strategies and market conditions. These agents operate in real-time, ensuring timely and accurate trade placements while adhering to user-defined parameters such as risk limits, position sizing, and strategy rules. The feature is designed to optimize trading efficiency, reduce manual intervention, and minimize errors, making it an essential tool for active traders and fund managers managing complex portfolios.

### Acceptance Criteria

- Execution agents must be able to place trades automatically based on predefined trading strategies and user-configured parameters.
- The agents should support multiple order types, including market orders, limit orders, stop-loss orders, and take-profit orders.
- Users must be able to configure risk management settings, such as maximum trade size, stop-loss thresholds, and position sizing rules, which the agents must strictly adhere to during trade execution.
- The agents should operate in real-time, ensuring minimal latency in trade placement and response to market conditions.
- Execution agents must integrate seamlessly with supported exchanges, utilizing standardized APIs for order placement and account management.
- The feature should include a monitoring dashboard where users can view agent activity, trade history, and performance metrics in real-time.
- Users must be able to pause, modify, or terminate agent operations at any time through the app interface.
- The agents should include error handling mechanisms to manage issues such as network connectivity problems or exchange-specific errors, providing clear feedback to users in case of failures.
- Execution agents must comply with user-defined risk thresholds, automatically halting operations if limits such as maximum drawdown or daily trading volume are exceeded.
- The feature should support dynamic adjustments to trading strategies based on real-time market volatility and conditions, ensuring optimal trade execution.
- Execution agents must log all actions and decisions for audit purposes, ensuring transparency and compliance with regulatory requirements.
- The feature should include robust security measures to protect sensitive data, such as API keys and trade information, using encryption and secure storage practices.
- The agents must be compatible with all supported devices and operating systems, ensuring consistent functionality across platforms.
- The feature should provide analytics and insights into agent performance, helping users refine their strategies and improve trading outcomes.

## Inter-agent communication framework with standardized data exchange

### Description

The 'Inter-agent communication framework with standardized data exchange' feature in the 'my-test-juan-0304' app enables seamless interaction between AI agents responsible for various trading tasks. This framework ensures that agents can efficiently share data, insights,

and decisions in a standardized format, facilitating collaboration and improving overall system performance. By enabling inter-agent communication, the app can handle complex trading scenarios, such as multi-strategy execution or cross-market analysis, with greater accuracy and speed. This feature is essential for creating a cohesive AI-driven trading ecosystem that adapts dynamically to market conditions and user preferences.

### Acceptance Criteria

- The communication framework must support standardized data formats (e.g., JSON, XML) to ensure compatibility and consistency across all AI agents.
- Agents must be able to exchange data in real-time, with minimal latency, to support high-frequency trading and rapid decision-making.
- The framework should include error handling mechanisms to manage data transmission failures or inconsistencies, ensuring system stability and reliability.
- Inter-agent communication must be secure, with encryption protocols in place to protect sensitive trading data and user information.
- The framework should support hierarchical and peer-to-peer communication models, allowing agents to collaborate based on task requirements and dependencies.
- Agents must be able to share insights, such as technical indicator analyses, strategy evaluations, and risk assessments, in a structured and actionable format.
- The system should include monitoring tools to track inter-agent communication performance, identify bottlenecks, and optimize data exchange processes.
- The framework must be scalable to accommodate additional agents or increased data volumes as the app grows and integrates more features.
- The communication framework should be compatible with the app's existing AI architecture and exchange integration systems, ensuring seamless operation across all components.
- The feature must be tested across different trading scenarios, including high-volatility markets, to ensure robustness and adaptability.
- The framework should allow for dynamic adjustments to communication protocols based on user preferences or market conditions, enhancing flexibility and customization.
- Documentation and developer tools must be provided to facilitate the integration of new agents into the communication framework.

## Feedback mechanism for refining agent decision-making based on trade outcomes

### Description

The 'Feedback mechanism for refining agent decision-making based on trade outcomes' feature in the 'my-test-juan-0304' app enables users to provide structured feedback on the performance of AI agents after trades are executed. This feedback loop allows the system to analyze trade outcomes, identify patterns, and refine agent algorithms for improved decision-making in future trades. By incorporating user insights and trade data, the feature ensures that AI agents adapt to changing market conditions and align with user-specific trading strategies. This feature is essential for enhancing the accuracy and reliability of automated trading decisions, ultimately improving portfolio performance.

### Acceptance Criteria

- Users must be able to access the feedback mechanism directly from the trade execution summary or analytics dashboard.
- Feedback options should include structured inputs such as rating scales (e.g., 1-5 stars), predefined categories (e.g., 'Execution Speed', 'Accuracy', 'Risk Management'), and open text fields for detailed comments.
- The system must analyze user feedback alongside trade data (e.g., profit/loss, execution time, market conditions) to identify actionable insights for agent refinement.
- AI agents should demonstrate measurable improvements in decision-making accuracy and performance over time, based on feedback analysis and algorithm updates.
- The feedback mechanism must include a visualization tool to show users how their feedback has influenced agent adjustments and improvements.
- Users should receive periodic updates on the impact of their feedback, including reports on agent performance changes and trade success rates.
- The feature must support real-time feedback submission and processing, ensuring timely updates to agent algorithms in response to market dynamics.
- Feedback data must be securely stored and anonymized to protect user privacy and comply with data protection regulations.
- The system should include error handling to manage incomplete or invalid feedback submissions, providing users with clear guidance on how to correct issues.
- The feedback mechanism must be compatible with all user roles that interact with AI agents, ensuring accessibility and usability across different devices and platforms.
- Integrate analytics to track user engagement with the feedback mechanism and measure its impact on agent performance improvements.
- Ensure the feedback interface adheres to WCAG (Web Content Accessibility Guidelines) standards, making it accessible to users with disabilities.

## Dynamic agent scaling to handle high market data volumes



## Description

The 'Dynamic agent scaling to handle high market data volumes' feature in the 'my-test-juan-0304' app ensures that the AI agents can efficiently process and respond to large volumes of market data in real-time. This feature dynamically adjusts the number of active agents based on the current data load, ensuring optimal performance and responsiveness. By leveraging scalable architecture, the system can handle market surges, high-frequency trading scenarios, and large datasets without compromising speed or accuracy. This is particularly beneficial for Active Traders and Fund Managers who rely on real-time insights and execution in volatile market conditions.

## Acceptance Criteria

- The system must automatically scale the number of active AI agents based on the volume of incoming market data, ensuring no delays in processing.
- Dynamic scaling should be seamless and should not interrupt ongoing agent operations or user interactions.
- The feature must support high-frequency trading scenarios, with the ability to handle data spikes during market surges without performance degradation.
- The system should provide real-time monitoring of agent activity and scaling status, accessible via a dashboard for Active Traders and Fund Managers.
- Scaling decisions must be based on predefined thresholds for data volume, latency, and system resource utilization, which can be customized by Fund Managers.
- The feature must include failover mechanisms to ensure system stability in case of unexpected scaling issues or resource constraints.
- Performance metrics, such as data processing speed, agent response time, and system resource usage, must be logged and available for review by users.
- The system should notify users (via in-app notifications or email) when significant scaling events occur, such as reaching maximum capacity or scaling down after a data surge.
- The feature must be tested across various market conditions, including extreme volatility, to ensure reliability and robustness.
- The dynamic scaling mechanism must comply with security and privacy standards, ensuring that sensitive data is not exposed during scaling operations.
- The feature should be optimized for cost-efficiency, ensuring that resource usage is proportional to the data load without unnecessary overhead.

## Real-time volatility monitoring and adjustment by agents

### Description

The 'Real-time volatility monitoring and adjustment by agents' feature in the 'my-test-juan-0304' app empowers users with advanced AI-driven capabilities to monitor market volatility in real-time and dynamically adjust trading strategies accordingly. This feature leverages sophisticated algorithms to analyze market conditions, detect sudden changes in volatility, and optimize trade execution parameters such as position sizing, stop-loss levels, and entry/exit points. By proactively responding to market fluctuations, this feature helps users mitigate risks and capitalize on opportunities, ensuring more informed and adaptive trading decisions.

### Acceptance Criteria

- The AI agents must continuously monitor market data streams to detect real-time volatility changes across supported trading pairs and exchanges.
- Volatility metrics should be displayed in a user-friendly dashboard, including indicators such as standard deviation, average true range (ATR), and implied volatility.
- The feature must provide automated adjustments to trading parameters, such as position sizing, stop-loss settings, and take-profit levels, based on current volatility levels.
- Users should have the ability to customize thresholds for volatility adjustments, allowing for personalized risk management strategies.
- The system must send real-time alerts to users via in-app notifications, email, or SMS when significant volatility changes are detected.
- The feature must integrate seamlessly with other AI agents, such as Execution Agents and Risk Management Agents, to ensure coordinated decision-making during volatile market conditions.
- Performance metrics, such as the effectiveness of volatility adjustments in improving trade outcomes, should be tracked and displayed in analytics dashboards.
- The feature must support backtesting capabilities, allowing users to simulate the impact of volatility adjustments on historical data.
- Error handling mechanisms must be implemented to ensure stable operation during high market data volumes or API failures, with fallback strategies in place.
- The feature must adhere to user privacy and data security standards, ensuring sensitive trading data is protected during real-time monitoring and adjustments.
- The system must be compatible with all supported exchanges and trading pairs, ensuring consistent functionality across different market environments.
- Accessibility features must be included to ensure the dashboard and alerts are usable by all users, including those with disabilities.

## AI-driven insights for portfolio optimization



## Description

The 'AI-driven insights for portfolio optimization' feature in 'my-test-juan-0304' app leverages advanced artificial intelligence algorithms to analyze market data, user trading patterns, and portfolio composition. It provides actionable recommendations to optimize portfolio performance, balancing risk and return based on individual user preferences and market conditions. This feature empowers users to make informed decisions by offering insights such as asset allocation strategies, diversification opportunities, and risk mitigation techniques. It is designed to cater to the unique needs of Active Traders, Passive Investors, and Fund Managers, ensuring tailored optimization strategies for each user role.

## Acceptance Criteria

- The feature must provide real-time AI-driven recommendations for portfolio optimization, including asset allocation, diversification, and risk management strategies.
- Recommendations should be personalized based on user preferences, trading history, and risk tolerance levels.
- The AI algorithms must analyze both historical and real-time market data to ensure accurate and timely insights.
- Users must be able to view detailed explanations for each recommendation, including the rationale and expected impact on portfolio performance.
- The feature should include visualizations such as charts and graphs to help users understand optimization suggestions and their potential outcomes.
- Active Traders should receive insights focused on short-term opportunities and high-frequency trading strategies, while Passive Investors should receive long-term, risk-adjusted recommendations. Fund Managers should receive advanced insights tailored to managing multiple portfolios.
- The feature must allow users to simulate the impact of implementing recommendations on their portfolio before making actual changes.
- Ensure compatibility with all major asset classes, including stocks, cryptocurrencies, ETFs, and bonds, to provide comprehensive optimization insights.
- The AI-driven insights must be accessible via both the web platform and mobile app, ensuring seamless user experience across devices.
- Include error handling mechanisms to manage potential issues with data retrieval or AI processing, providing users with clear feedback in case of errors.
- The feature must adhere to data privacy and security standards, ensuring that sensitive user information is protected during analysis and recommendation generation.
- Integrate analytics to track user engagement with the feature and measure the effectiveness of recommendations in improving portfolio performance.
- Ensure the feature is tested across different user roles and devices to guarantee consistent functionality and usability.
- The interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Cross-agent collaboration for complex trading scenarios

### Description

The 'Cross-agent collaboration for complex trading scenarios' feature in 'my-test-juan-0304' enables multiple AI agents to work together seamlessly to tackle sophisticated trading challenges. By leveraging inter-agent communication and shared data analysis, this feature allows agents to combine their specialized capabilities—such as indicator analysis, strategy evaluation, and risk assessment—to optimize trading decisions in real-time. This collaborative approach is particularly beneficial for handling multi-layered trading strategies, cross-market arbitrage, and high-frequency trading scenarios. The feature enhances decision-making accuracy, adaptability, and overall trading performance, making it a valuable tool for advanced traders and fund managers.

### Acceptance Criteria

- AI agents must be able to communicate and share data in real-time using a standardized inter-agent communication framework.
- Agents must collaborate to analyze complex trading scenarios, such as multi-market arbitrage or high-frequency trading, and provide actionable insights to users.
- The collaboration process should dynamically allocate tasks to agents based on their specialized functions (e.g., indicator analysis, risk assessment, execution management).
- The system must demonstrate high performance and low latency in inter-agent communication, ensuring timely decision-making during volatile market conditions.
- Users must be able to view a detailed breakdown of how agents collaborated to arrive at a trading decision, including the roles and contributions of each agent.
- The feature must support scalability, allowing additional agents to be integrated into the collaboration framework as needed without compromising performance.
- Cross-agent collaboration must include error handling mechanisms to address potential communication failures or conflicting agent recommendations.
- The feature must prioritize user privacy and data security, ensuring that sensitive trading data is protected during inter-agent communication.
- Provide analytics and reporting tools to track the effectiveness of cross-agent collaboration, including metrics such as trade success rates and decision accuracy.

- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Ensure the collaboration framework adheres to relevant financial regulations and compliance standards, particularly for fund managers handling client portfolios.
- Include interactive tutorials or documentation to help users understand how cross-agent collaboration works and how to leverage it effectively.

# Exchange Integration

## Standardized API integration framework for market data retrieval and order placement

### Description

The 'Standardized API integration framework for market data retrieval and order placement' feature in the 'my-test-juan-0304' app provides a unified and efficient mechanism for connecting to various cryptocurrency exchanges. This framework ensures seamless retrieval of real-time market data and facilitates secure and accurate order placement across supported exchanges. By standardizing the integration process, the feature reduces complexity, enhances reliability, and ensures compatibility with multiple exchanges. It is designed to support high-frequency trading, portfolio management, and strategic decision-making, catering to the diverse needs of Active Traders, Passive Investors, and Fund Managers.

### Acceptance Criteria

- The API integration framework must support real-time market data retrieval, including price, volume, and order book information, from all integrated exchanges.
- Users must be able to place, modify, and cancel orders (e.g., market, limit, stop-loss) through the framework with minimal latency and high accuracy.
- The framework must include standardized data formats for market data and order placement to ensure consistency across exchanges.
- Integration with major exchanges (e.g., Binance, Coinbase Pro, Kraken, FTX) must be fully functional, with a roadmap for additional exchange integrations within 12 months.
- The framework must comply with rate-limiting policies of each exchange to prevent API call restrictions or bans.
- Error handling mechanisms must be implemented to manage exchange-specific issues, such as downtime or invalid API responses, with clear feedback provided to users.
- The framework must support multiple trading pairs across exchanges, allowing users to trade a wide range of cryptocurrencies and tokens.
- Real-time account balance monitoring must be integrated, enabling users to view their holdings and available funds across all connected exchanges.
- The framework must include secure storage and handling of exchange API keys, utilizing AES-256 encryption to protect sensitive user data.
- Performance metrics, such as API response time and order execution speed, must be monitored and optimized regularly to ensure high performance.
- The framework must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The feature must adhere to relevant cryptocurrency regulations and compliance standards, ensuring legal and ethical usage.
- Users must be able to view detailed logs of API interactions, including market data retrieval and order placement, for transparency and troubleshooting.
- The framework must support scalability to handle high market data volumes and large numbers of concurrent users without performance degradation.

## Exchange-specific adapters for unique features and limitations

### Description

The 'Exchange-specific adapters for unique features and limitations' feature in the 'my-test-juan-0304' app ensures seamless integration with various cryptocurrency exchanges by accommodating their unique functionalities, constraints, and APIs. These adapters act as tailored bridges between the app and individual exchanges, enabling users to leverage exchange-specific features such as advanced order types, fee structures, and trading limits. By addressing the nuances of each exchange, this feature enhances trading efficiency, reduces errors, and provides a consistent user experience across multiple platforms.

### Acceptance Criteria

- The app must support exchange-specific adapters that handle unique features, such as advanced order types (e.g., stop-limit, trailing stop), fee structures, and trading limits.
- Adapters must be designed to accommodate differences in API protocols, ensuring compatibility with each exchange's requirements.
- The feature must include error handling mechanisms to manage exchange-specific issues, such as rate limits, downtime, or API changes, and provide clear feedback to users.
- Adapters must support real-time synchronization of account balances, open orders, and trade history for each integrated exchange.
- Users must be able to access exchange-specific features directly through the app interface without needing to interact with the exchange's native platform.
- The app must provide documentation or tooltips explaining the unique features and limitations of each supported exchange, ensuring users understand how to utilize them effectively.

- Adapters must be tested for reliability and performance across all supported exchanges, ensuring quick response times and minimal latency during data retrieval and order placement.
- The feature must include a roadmap for adding support for new exchanges, with a focus on scalability and modularity to simplify future integrations.
- Adapters must comply with security best practices, including encryption of sensitive data and secure handling of API keys, to protect user accounts and information.
- The feature must be compatible with the app's risk management tools, ensuring that exchange-specific constraints (e.g., maximum trade size) are factored into risk calculations.

## Integration with major exchanges (Binance, Coinbase Pro, Kraken, FTX)

### Description

The 'Integration with major exchanges' feature in the 'my-test-juan-0304' app enables seamless connectivity with leading cryptocurrency exchanges, including Binance, Coinbase Pro, Kraken, and FTX. This integration allows users to retrieve real-time market data, monitor account balances, and execute trades directly from the app. By leveraging standardized APIs, the feature ensures reliable and efficient communication with these exchanges, providing users with a unified platform for managing their trading activities. This feature is essential for empowering users to make informed decisions and execute strategies across multiple exchanges without the need for separate tools.

### Acceptance Criteria

- The app must support real-time market data retrieval from Binance, Coinbase Pro, Kraken, and FTX, including price updates, order book data, and trade history.
- Users must be able to link their exchange accounts securely using API keys, with encryption (AES-256) to protect sensitive information.
- The integration must allow users to monitor account balances, including available funds, locked funds, and portfolio value across all supported exchanges.
- Users must be able to place, modify, and cancel orders directly from the app, with support for market, limit, and stop orders.
- The app must handle exchange-specific features and limitations, such as rate limits, trading pairs, and order types, ensuring compatibility with each platform.
- Error handling mechanisms must be implemented to manage issues such as API failures, rate limit breaches, or invalid credentials, providing clear feedback to users.
- The integration must support multiple trading pairs across all exchanges, allowing users to trade a wide range of cryptocurrencies.
- Real-time synchronization of data between the app and exchanges must be ensured, with minimal latency for accurate decision-making.
- The app must comply with the terms of service and API usage policies of each exchange, avoiding violations that could disrupt functionality.
- Users must be able to view detailed transaction histories for trades executed through the app, including timestamps, trade amounts, and fees.
- The feature must include a roadmap for adding support for additional exchanges, with a target of integrating 10+ exchanges within 12 months.
- The integration must be tested across different devices and operating systems to ensure consistent performance and reliability.
- The feature must include analytics for tracking user activity across exchanges, providing insights into trading behavior and performance.

## Roadmap for additional exchange integrations (10+ exchanges within 12 months)

### Description

The 'Roadmap for additional exchange integrations' feature in the 'my-test-juan-0304' app outlines a strategic plan to expand the app's compatibility with over 10 new cryptocurrency exchanges within the next 12 months. This feature is designed to enhance trading opportunities for users by providing access to a broader range of markets, trading pairs, and liquidity pools. The roadmap includes detailed milestones for integration, ensuring seamless connectivity, compliance with exchange-specific requirements, and optimized performance. By increasing the number of supported exchanges, the app aims to cater to diverse trading preferences and strategies, empowering users to maximize their trading potential.

### Acceptance Criteria

- The roadmap must clearly define milestones for integrating 10+ new exchanges within 12 months, including timelines, technical requirements, and testing phases.
- Each new exchange integration must support core functionalities such as market data retrieval, order placement, account balance monitoring, and trading pair support.
- The app must ensure compliance with exchange-specific API rate limits and error handling mechanisms to maintain stable performance.
- All new integrations must undergo rigorous testing to ensure compatibility with the app's existing features and user workflows.

- The roadmap must prioritize exchanges based on user demand, market relevance, and trading volume, ensuring maximum value for users.
- Integration with new exchanges must include support for both spot and derivative trading, where applicable, to cater to diverse trading strategies.
- The app must provide regular updates to users about the progress of exchange integrations, including announcements for newly added exchanges.
- Newly integrated exchanges must be accessible to all user roles (Active Trader, Passive Investor, Fund Manager) without restrictions, ensuring equal opportunities for trading and portfolio management.
- The app must include documentation and tutorials for each new exchange integration, helping users understand unique features and limitations of the exchanges.
- Performance metrics for new integrations, such as response times and data accuracy, must be monitored and optimized to ensure a seamless user experience.
- The roadmap must include contingency plans for potential delays or issues with exchange integrations, ensuring transparency and accountability.
- The app must adhere to relevant regulatory requirements for each new exchange integration, ensuring compliance and user trust.

## Rate limiting compliance for API calls

### Description

The 'Rate limiting compliance for API calls' feature in the 'my-test-juan-0304' app ensures that all interactions with integrated exchanges adhere to their respective API rate limits. This feature is critical for maintaining smooth and uninterrupted communication with exchanges, preventing service disruptions caused by exceeding rate limits. It dynamically monitors API usage and implements throttling mechanisms to regulate the frequency of requests. By ensuring compliance, this feature protects the app from potential penalties or bans imposed by exchanges and enhances overall system reliability.

### Acceptance Criteria

- The app must dynamically monitor API usage for each integrated exchange and track the number of requests made within the allowed time frame.
- Implement a throttling mechanism to automatically regulate the frequency of API calls, ensuring compliance with rate limits set by each exchange.
- Provide real-time feedback to users when API rate limits are approaching, including notifications or warnings within the app interface.
- Ensure that API calls are prioritized based on their importance, such as trade execution requests taking precedence over data retrieval requests, to optimize system performance.
- Include a fallback mechanism to queue and retry API calls that exceed rate limits, ensuring that critical operations are not lost or disrupted.
- Support exchange-specific rate limit configurations, allowing the app to adapt to varying limits across different exchanges.
- Provide detailed logs and analytics for API usage, enabling users and administrators to monitor compliance and identify potential issues.
- Ensure that the rate limiting mechanism is compatible with high-frequency trading scenarios, maintaining performance without exceeding limits.
- Include error handling to gracefully manage scenarios where rate limits are exceeded, providing clear feedback to users and preventing app crashes.
- Test the rate limiting feature across all integrated exchanges to ensure consistent behavior and compliance with their respective requirements.
- Ensure that the rate limiting mechanism is scalable to handle increased API call volumes as the app integrates with additional exchanges.
- Integrate analytics to track API usage patterns and optimize the rate limiting algorithm for improved efficiency and user experience.

## Error handling for exchange-specific issues

### Description

The 'Error handling for exchange-specific issues' feature in the 'my-test-juan-0304' app ensures robust management of errors arising from exchange-specific limitations or failures. This feature is designed to identify, categorize, and respond to issues such as API rate limits, connectivity problems, invalid order parameters, or exchange downtime. By providing clear error messages, actionable recommendations, and fallback mechanisms, it minimizes disruptions to trading activities and enhances user confidence in the platform's reliability. This feature is critical for maintaining seamless operations across integrated exchanges and ensuring user satisfaction.

### Acceptance Criteria

- The app must detect and categorize exchange-specific errors, including API rate limits, connectivity issues, invalid order parameters, and exchange downtime.
- Clear and user-friendly error messages must be displayed, providing detailed information about the issue and potential solutions.
- The app should implement retry mechanisms for temporary errors, such as network connectivity issues or rate limit breaches, with configurable retry intervals.

- Fallback mechanisms must be in place to handle prolonged exchange downtime, such as switching to alternative exchanges or notifying users of the issue.
- Users must be notified of critical errors via in-app alerts, email, or SMS, depending on their notification preferences.
- The app should log all exchange-specific errors for auditing and troubleshooting purposes, ensuring compliance with regulatory requirements.
- Error handling must be optimized for performance, ensuring that the app remains responsive even during high error volumes or exchange outages.
- The app should provide actionable recommendations for users to resolve errors, such as adjusting API call frequency or verifying order parameters.
- Integration with analytics tools must be included to track error patterns and identify opportunities for improving exchange integration reliability.
- The error handling system must be tested across all integrated exchanges to ensure consistent functionality and compatibility.
- The feature must adhere to security best practices, ensuring that sensitive user data is not exposed during error handling processes.

## Real-time account balance monitoring

### Description

The 'Real-time account balance monitoring' feature in the 'my-test-juan-0304' app provides users with up-to-date information on their account balances across integrated exchanges. This feature ensures that users can make informed trading decisions by displaying accurate and timely data on available funds, locked balances, and margin availability. It supports multiple trading pairs and accounts, offering a consolidated view of balances for seamless portfolio management. The feature is designed to enhance user experience by providing transparency and reducing the risk of over-leveraging or insufficient funds during trade execution.

### Acceptance Criteria

- The account balance data must be updated in real-time, reflecting changes due to executed trades, deposits, withdrawals, or market fluctuations.
- Users must be able to view balances for all integrated exchanges in a consolidated dashboard, with the option to filter by individual exchange or trading pair.
- The feature must display detailed balance information, including available funds, locked balances (e.g., for open orders), and margin availability for leveraged trading accounts.
- Balances must be presented in both the native currency of the trading pair and the user's preferred base currency (e.g., USD, EUR, BTC), with automatic conversion using real-time exchange rates.
- The app must provide visual indicators (e.g., color-coded alerts) for low balances or margin calls, helping users avoid potential trading risks.
- Users must be able to refresh the balance data manually, in addition to the automatic real-time updates, ensuring accuracy during high-frequency trading scenarios.
- The feature must include error handling for scenarios where exchange APIs fail or experience delays, displaying a clear message to users and retrying the data fetch automatically.
- The account balance monitoring must be optimized for performance, ensuring quick response times even when handling large datasets across multiple exchanges and accounts.
- The feature must adhere to security best practices, ensuring that sensitive account information is encrypted and protected during data retrieval and display.
- The interface must be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards to accommodate users with disabilities.
- The feature must be compatible with the mobile app, providing seamless access to real-time balance data on both desktop and mobile platforms.
- Integrate analytics to track user interactions with the balance monitoring feature, providing insights for future improvements and optimizations.

## Support for multiple trading pairs across exchanges

### Description

The 'Support for multiple trading pairs across exchanges' feature in the 'my-test-juan-0304' app enables users to trade a wide variety of cryptocurrency pairs seamlessly across integrated exchanges. This feature ensures that users can access and manage trading pairs for major cryptocurrencies, altcoins, and stablecoins, providing flexibility and diversity in trading opportunities. It includes real-time updates on available trading pairs, compatibility with exchange-specific offerings, and efficient handling of pair-specific data such as price, volume, and liquidity. This feature is essential for empowering users to execute trades across multiple markets and optimize their trading strategies.

### Acceptance Criteria

- The app must display a comprehensive list of available trading pairs for each integrated exchange, updated in real-time.
- Users must be able to search, filter, and sort trading pairs based on criteria such as asset type, volume, liquidity, and price.
- The feature must support trading pairs for major cryptocurrencies (e.g., BTC, ETH), altcoins, and stablecoins, ensuring broad market coverage.

- Users must be able to select and trade any supported pair directly from the app interface, with seamless integration into the order placement workflow.
- The app must handle exchange-specific limitations or restrictions on trading pairs, providing clear feedback to users when certain pairs are unavailable.
- Real-time data for trading pairs, including price, volume, and liquidity, must be displayed accurately and updated frequently to ensure informed decision-making.
- The feature must support cross-exchange comparisons of trading pairs, enabling users to identify arbitrage opportunities and optimize their trades.
- Error handling must be implemented to manage issues such as unavailable trading pairs, API failures, or network connectivity problems, ensuring stability and user feedback.
- The feature must be compatible with all integrated exchanges, including Binance, Coinbase Pro, Kraken, and others, with a roadmap for future exchange integrations.
- Performance metrics such as response time and data accuracy must be monitored regularly to ensure high performance and reliability.
- The feature must adhere to user privacy and data security standards, ensuring sensitive information related to trading pairs is protected.
- The interface for managing trading pairs must be intuitive and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards for usability.

## On-chain transaction data integration for fundamental analysis

### Description

The 'On-chain transaction data integration for fundamental analysis' feature in the 'my-test-juan-0304' app enables users to leverage blockchain data for informed trading decisions. By integrating on-chain transaction data, this feature provides insights into market trends, token movements, and wallet activity, offering a deeper understanding of asset fundamentals. Users can analyze metrics such as transaction volume, active addresses, token distribution, and large wallet movements to assess the health and potential of cryptocurrencies. This feature is essential for traders and fund managers seeking to incorporate fundamental analysis into their strategies, complementing technical and sentiment-based approaches.

### Acceptance Criteria

- The app must retrieve and display on-chain transaction data from major blockchains, including Ethereum, Bitcoin, and other supported networks.
- Users must be able to view key on-chain metrics such as transaction volume, active addresses, token distribution, and large wallet movements in an intuitive dashboard.
- The feature should provide historical on-chain data for trend analysis, allowing users to compare current metrics with past performance.
- Data visualization tools, such as charts and graphs, must be included to help users interpret on-chain metrics effectively.
- The app should allow users to filter on-chain data by specific tokens, timeframes, and transaction types for focused analysis.
- Integration with blockchain explorers (e.g., Etherscan, Blockchain.com) must be implemented to provide detailed transaction-level insights.
- The feature must support real-time updates for on-chain data, ensuring users have access to the latest information for decision-making.
- On-chain data must be integrated seamlessly with other analytics features in the app, such as predictive analytics and portfolio optimization tools.
- The app should include alerts and notifications for significant on-chain events, such as large token transfers or spikes in transaction volume.
- Ensure high performance and reliability in handling large volumes of blockchain data, with minimal latency and downtime.
- The feature must adhere to data privacy and security standards, ensuring sensitive blockchain data is handled securely and responsibly.
- Provide educational resources or tooltips within the app to help users understand and utilize on-chain metrics effectively.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Ensure accessibility of the on-chain data dashboard, adhering to WCAG (Web Content Accessibility Guidelines) standards for usability by all users.

## Cross-exchange arbitrage capabilities

### Description

The 'Cross-exchange arbitrage capabilities' feature in the 'my-test-juan-0304' app enables users to identify and execute arbitrage opportunities across multiple cryptocurrency exchanges. By leveraging real-time market data, this feature allows users to capitalize on price discrepancies for the same trading pair across different exchanges. It provides tools for automated trade execution, ensuring swift and efficient transactions to maximize profits. This feature is particularly valuable for users seeking to optimize their trading strategies and generate consistent returns in volatile markets.

## Acceptance Criteria

- The app must support integration with multiple exchanges, including Binance, Coinbase Pro, Kraken, and FTX, with plans for additional integrations as outlined in the roadmap.
- Users must be able to view real-time price data for the same trading pair across all integrated exchanges, displayed in an intuitive and easily navigable interface.
- The feature must include an automated arbitrage execution system that can place buy and sell orders across exchanges simultaneously to capitalize on price discrepancies.
- The app should provide users with a profitability calculator that factors in trading fees, withdrawal fees, and network transaction costs to ensure accurate profit estimation before executing arbitrage trades.
- Users must be able to set custom thresholds for price discrepancies, enabling the system to trigger arbitrage trades only when the potential profit meets or exceeds their defined criteria.
- The feature must include error handling mechanisms to address issues such as failed orders, network delays, or exchange-specific restrictions, ensuring stability and reliability during arbitrage execution.
- The app should provide detailed logs and analytics for all arbitrage trades, including timestamps, executed prices, fees incurred, and net profit, allowing users to track performance and refine their strategies.
- The feature must comply with rate-limiting requirements for API calls across all integrated exchanges to avoid disruptions or bans from exchange platforms.
- Users must be able to enable or disable the automated arbitrage system at any time, with manual override options for greater control over trade execution.
- The app should include a risk management system to prevent excessive exposure, such as limiting the maximum amount allocated for arbitrage trades or pausing operations during extreme market volatility.
- The feature must demonstrate high performance and quick response times, ensuring that arbitrage opportunities are identified and executed before market conditions change.
- The app should provide educational resources, such as tutorials and guides, to help users understand how cross-exchange arbitrage works and how to use the feature effectively.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistent functionality.
- The feature must adhere to security best practices, including encryption of sensitive data and secure handling of exchange API keys, to protect user assets and information.



# Trading Strategies

## Implementation of core trading strategies (e.g., scalping, swing trading, trend following)

### Description

The 'Implementation of core trading strategies' feature in the 'my-test-juan-0304' app provides users with a suite of pre-configured trading strategies, including scalping, swing trading, and trend following. These strategies are designed to cater to different trading styles and risk appetites, enabling users to optimize their trading performance. The feature offers detailed strategy descriptions, adjustable parameters, and real-time analytics to help users make informed decisions. It is essential for empowering users with proven methodologies to navigate the complexities of financial markets effectively.

### Acceptance Criteria

- The app must provide a library of pre-configured trading strategies, including scalping, swing trading, and trend following, with detailed descriptions of each strategy's objectives and mechanics.
- Users must be able to select and activate a trading strategy directly from their dashboard, with clear instructions on how to use it.
- Each strategy must include adjustable parameters (e.g., stop-loss levels, take-profit targets, position sizing) to allow users to customize the strategy to their preferences and risk tolerance.
- The app should provide real-time performance analytics for each strategy, including metrics such as win rate, profit factor, and drawdown, enabling users to evaluate effectiveness.
- Strategies must be compatible with both live trading and simulation mode, allowing users to test strategies with real-time market data or historical data before deploying them in live markets.
- The app must include a tutorial or guide for each strategy, explaining its use case, advantages, and potential risks, ensuring users understand how to apply it effectively.
- Users must be able to combine multiple strategies within their portfolio, with the app providing insights into how these strategies interact and impact overall performance.
- The feature must support integration with major exchanges (e.g., Binance, Coinbase Pro, Kraken) to ensure seamless execution of trades based on the selected strategy.
- The app should provide notifications and alerts for strategy-related events, such as when a strategy hits a stop-loss or take-profit target, ensuring users stay informed in real-time.
- The feature must include error handling to manage issues such as failed trade executions or discrepancies in market data, providing users with clear feedback and resolution steps.
- The app must ensure that all strategy-related data and analytics are securely stored and encrypted, adhering to industry standards for data protection.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and performance.
- The strategy interface must be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards to accommodate users with disabilities.

## Strategy customization interface with adjustable parameters

### Description

The 'Strategy customization interface with adjustable parameters' feature in the 'my-test-juan-0304' app empowers users to tailor trading strategies to their specific needs and preferences. This interface provides an intuitive and flexible platform for adjusting key parameters such as entry and exit conditions, risk tolerance, position sizing, and timeframes. By enabling users to fine-tune strategies, the feature enhances their ability to optimize performance and align trading activities with their goals. It is designed to cater to advanced users who require granular control over their trading strategies, while maintaining ease of use for efficient customization.

### Acceptance Criteria

- The strategy customization interface must be accessible from the main dashboard and clearly labeled for easy navigation.
- Users must be able to adjust key parameters such as entry and exit conditions, stop-loss and take-profit levels, position sizing, and timeframes.
- The interface should provide real-time previews of how parameter adjustments impact the strategy's performance metrics, including potential profit, risk exposure, and win rate.
- Include pre-configured templates for common trading strategies (e.g., scalping, swing trading, trend following) that users can modify to suit their needs.
- Provide a user-friendly interface with sliders, dropdowns, and input fields for parameter adjustments, ensuring simplicity and accessibility.
- Enable users to save customized strategies for future use, with options to name and categorize them for easy retrieval.
- Allow users to backtest customized strategies using historical data to evaluate their effectiveness before live implementation.
- Include error handling to validate user inputs and prevent invalid or conflicting parameter configurations.
- Ensure compatibility with AI-driven insights, allowing users to incorporate AI recommendations into their customized strategies.
- Provide detailed documentation and interactive tutorials to guide users through the customization process.
- The interface must demonstrate high performance and responsiveness, ensuring smooth adjustments without delays.

- Ensure the feature is compatible across all devices, including desktop and mobile platforms, with a consistent user experience.
- Adhere to WCAG (Web Content Accessibility Guidelines) standards to ensure usability for all users, including those with disabilities.
- Integrate analytics to track user behavior and preferences within the customization interface, enabling continuous improvement of the feature.

## Real-time and historical performance analytics for strategies

### Description

The 'Real-time and historical performance analytics for strategies' feature in the 'my-test-juan-0304' app provides users with comprehensive insights into the effectiveness of their trading strategies. This feature enables users to monitor the real-time performance of active strategies and analyze historical data to evaluate past outcomes. It includes detailed metrics such as win rate, profit factor, drawdown, and return on investment (ROI), empowering users to make informed decisions about strategy adjustments and optimizations. By offering both real-time and historical analytics, this feature supports users in identifying trends, assessing risk, and improving their overall trading performance.

### Acceptance Criteria

- The feature must provide real-time performance metrics for active trading strategies, including win rate, profit factor, drawdown, and ROI.
- Historical performance analytics must be accessible, allowing users to review past strategy outcomes across multiple timeframes (e.g., daily, weekly, monthly).
- The analytics dashboard must be visually intuitive, featuring charts, graphs, and tables for easy interpretation of data.
- Users must be able to filter and sort performance data by specific criteria, such as strategy type, timeframe, or trading pair.
- The feature must support exporting performance analytics reports in common formats (e.g., PDF, CSV) for offline review and compliance purposes.
- Real-time analytics must update dynamically without requiring manual refresh, ensuring users have the most current data available.
- Historical data must be stored securely and remain accessible for a minimum of 12 months, with options for extended storage for premium users.
- The analytics must include benchmarks for comparison, such as market indices or manual trading performance, to provide context for strategy evaluation.
- The feature must include predictive analytics capabilities, offering insights into potential future performance based on historical trends and current market conditions.
- Users must be able to customize the analytics dashboard to display preferred metrics and visualizations, enhancing usability for different trading styles.
- The feature must demonstrate high performance and responsiveness, ensuring quick data retrieval and visualization even for large datasets.
- Error handling must be implemented to manage potential issues with data retrieval or processing, providing clear feedback to users in case of failures.
- The analytics must adhere to data privacy and security standards, ensuring sensitive trading information is protected and not exposed.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The analytics interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Backtesting engine using historical data across multiple timeframes

### Description

The 'Backtesting engine using historical data across multiple timeframes' feature in the 'my-test-juan-0304' app allows users to evaluate the performance of trading strategies by simulating trades using historical market data. This feature enables users to test strategies across various timeframes, such as intraday, daily, weekly, or monthly, providing insights into their effectiveness under different market conditions. By leveraging this tool, users can refine their strategies, identify potential weaknesses, and optimize their trading approaches before deploying them in live markets. The backtesting engine is designed to deliver accurate and actionable results, empowering users to make informed decisions and improve their overall trading performance.

### Acceptance Criteria

- The backtesting engine must support multiple timeframes, including intraday, daily, weekly, and monthly, allowing users to test strategies across diverse market conditions.
- Users must be able to select specific historical data periods for backtesting, with options to filter by date range, asset type, and trading pair.
- The engine should provide detailed performance metrics, including win rate, profit factor, drawdown, and average trade duration, to help users evaluate strategy effectiveness.
- Backtesting results must be presented in an intuitive and visually appealing format, such as charts, graphs, and tables, for easy interpretation and analysis.
- The feature must allow users to compare the performance of multiple strategies side-by-side, enabling them to identify the most effective approach.

- The backtesting engine should integrate with real-time market data to ensure accurate simulations and reflect current market conditions.
- Users must be able to save and export backtesting results for further analysis or sharing with team members and collaborators.
- The engine should include error handling to manage issues such as incomplete historical data or invalid strategy parameters, providing clear feedback to users about the problem.
- Performance optimization must ensure that backtesting simulations are completed quickly, even when processing large datasets or complex strategies.
- The feature must adhere to user privacy and data security standards, ensuring that sensitive information is not exposed during the backtesting process.
- The backtesting engine should be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Interactive tutorials or tooltips must be available to guide users through the backtesting process, ensuring accessibility for both novice and experienced traders.
- The feature must support integration with other tools in the app, such as the strategy customization interface and AI-driven strategy recommendations, for seamless workflow.

## Simulation mode for paper trading with real-time market data

### Description

The 'Simulation mode for paper trading with real-time market data' feature in the 'my-test-juan-0304' app allows users to practice trading strategies in a risk-free environment using live market data. This feature is designed to help users refine their trading skills, test strategies, and gain confidence without the financial risk of actual trades. The simulation mode replicates real-world trading conditions, including market volatility, order execution, and price movements, providing an authentic experience. It is an essential tool for both novice and experienced traders to evaluate the effectiveness of their strategies and make informed decisions before committing real funds.

### Acceptance Criteria

- The simulation mode must be accessible from the main dashboard under the 'Trading Strategies' section.
- Users must be able to toggle between simulation mode and live trading mode seamlessly.
- The simulation must use real-time market data from integrated exchanges to ensure accuracy and relevance.
- Users must be able to execute simulated trades, including market orders, limit orders, and stop-loss orders, with the same interface as live trading.
- The simulation mode must provide detailed performance metrics, including profit/loss, win rate, drawdown, and trade history, to help users analyze their strategies.
- Simulated trades must not affect actual account balances or portfolios, ensuring a completely risk-free environment.
- The feature must support all trading pairs available in the app, allowing users to test strategies across different markets.
- Users must be able to customize simulation parameters, such as initial virtual balance and leverage settings, to match their desired trading conditions.
- The app must display a clear indicator when users are in simulation mode to avoid confusion with live trading.
- Simulation results must be stored and accessible for future reference, enabling users to track their progress and refine strategies over time.
- The feature must demonstrate high performance and responsiveness, ensuring real-time updates and smooth user experience during simulations.
- Include error handling to manage potential issues with market data retrieval or simulation execution, providing feedback to users about any disruptions.
- The simulation mode must be compatible with mobile and desktop versions of the app, ensuring consistent functionality across devices.
- Ensure the simulation interface adheres to accessibility standards, making it usable for all users, including those with disabilities.

## Dynamic position sizing based on market volatility

### Description

The 'Dynamic position sizing based on market volatility' feature in the 'my-test-juan-0304' app enables users to optimize their trade sizes by automatically adjusting position sizes based on real-time market volatility. This feature leverages advanced algorithms to assess market conditions, ensuring that users can manage risk effectively while maximizing potential returns. By dynamically scaling position sizes, traders and fund managers can adapt to changing market environments, reducing exposure during high volatility and capitalizing on opportunities during stable conditions. This feature is essential for maintaining a balanced risk-reward profile and improving overall trading performance.

### Acceptance Criteria

- The feature must provide real-time analysis of market volatility using metrics such as Average True Range (ATR), Bollinger Bands, or implied volatility.
- Position sizes must be automatically adjusted based on predefined rules or user-configured parameters, such as risk tolerance and portfolio size.
- Users must be able to set maximum and minimum position size limits to ensure control over trade exposure.

- The feature should include a visual representation of volatility metrics and position size adjustments within the trading dashboard for transparency.
- Dynamic position sizing must integrate seamlessly with other trading strategies and tools within the app, ensuring compatibility and ease of use.
- The feature must support backtesting capabilities, allowing users to evaluate the effectiveness of dynamic position sizing using historical data.
- Provide notifications or alerts to users when significant changes in position sizes occur due to market volatility shifts.
- Ensure high performance and responsiveness, with position size adjustments occurring in real-time without delays.
- Include error handling mechanisms to manage potential issues, such as data retrieval failures or extreme market conditions, ensuring stability and reliability.
- The feature must adhere to user privacy and data security standards, ensuring sensitive trading data is protected.
- Offer customization options for advanced users, such as the ability to define volatility thresholds and scaling factors.
- Ensure compatibility across different devices and operating systems, including mobile and desktop versions of the app.
- Provide an interactive tutorial or help section to guide users on how to configure and utilize dynamic position sizing effectively.

## Portfolio optimization tools for risk-adjusted returns

### Description

The 'Portfolio optimization tools for risk-adjusted returns' feature in the 'my-test-juan-0304' app provides users with advanced tools to optimize their investment portfolios by balancing risk and return. Leveraging AI-driven insights and analytics, this feature enables users to identify the most efficient allocation of assets based on their risk tolerance, investment goals, and market conditions. It incorporates modern portfolio theory principles and real-time data to suggest adjustments that maximize returns while minimizing exposure to risk. This feature is essential for empowering users to make informed decisions and achieve sustainable growth in their investments.

### Acceptance Criteria

- The portfolio optimization tools must be accessible from the main dashboard under the 'Trading Strategies' section.
- Users must be able to input their risk tolerance levels, investment goals, and preferred asset classes to customize optimization recommendations.
- The feature should utilize AI-driven algorithms to analyze historical and real-time market data, providing actionable insights for portfolio adjustments.
- Optimization recommendations must include detailed metrics such as expected returns, risk levels (e.g., standard deviation), and Sharpe ratios to help users evaluate suggested changes.
- Users must be able to visualize their current portfolio allocation and compare it with the optimized allocation using interactive charts and graphs.
- The feature should support multiple asset types, including cryptocurrencies, stocks, and ETFs, ensuring comprehensive portfolio analysis.
- Provide users with the ability to simulate the impact of optimization recommendations on their portfolio performance using historical data and predictive analytics.
- Include a 'What-If' analysis tool to allow users to test different scenarios and assess the impact of market changes on their optimized portfolio.
- Ensure the optimization process adheres to user-defined constraints, such as maximum exposure to specific assets or industries.
- The feature must offer real-time alerts and notifications for significant market changes that may impact the optimized portfolio, enabling users to take timely action.
- Integrate a risk visualization dashboard to help users understand their current exposure and how optimization recommendations reduce risk.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.
- The optimization tools must prioritize user privacy and data security, ensuring sensitive financial information is protected.
- Provide an interactive tutorial to guide users through the portfolio optimization process, making the feature accessible to both novice and experienced investors.

## AI-driven strategy recommendations based on market conditions

### Description

The 'AI-driven strategy recommendations based on market conditions' feature in the 'my-test-juan-0304' app leverages advanced artificial intelligence algorithms to analyze real-time and historical market data, providing users with tailored trading strategies optimized for current market conditions. This feature empowers users to make informed decisions by offering actionable insights, such as suggested entry and exit points, risk-adjusted strategies, and portfolio adjustments. It adapts dynamically to market volatility, trends, and user preferences, ensuring that recommendations align with individual trading goals and risk tolerance. By integrating AI-driven insights, this feature enhances trading efficiency and helps users maximize their returns while minimizing risks.

### Acceptance Criteria

- The feature must analyze real-time and historical market data to generate strategy recommendations tailored to current market conditions.

- Recommendations should include actionable insights such as entry and exit points, risk-adjusted strategies, and portfolio adjustments.
- The AI algorithms must dynamically adapt to market volatility, trends, and user preferences to ensure relevance and accuracy.
- Users must be able to customize the recommendations based on their trading goals, risk tolerance, and preferred trading strategies.
- The feature should provide visualizations (e.g., charts, graphs) to help users understand the rationale behind the recommendations.
- Recommendations must be categorized by trading strategy types (e.g., scalping, swing trading, trend following) for easy navigation and selection.
- The feature must support multi-asset analysis, including cryptocurrencies, stocks, and other tradable assets, depending on the user's portfolio.
- Users should receive notifications or alerts for new recommendations, especially during significant market events or changes.
- The AI-driven recommendations must demonstrate high performance and quick response times, ensuring timely delivery of insights during fast-moving markets.
- Include error handling to manage potential failures in data retrieval or AI processing, providing fallback options or explanations to users.
- The feature must prioritize user privacy and data security, ensuring that sensitive trading data is not exposed or misused.
- Integrate analytics to track the effectiveness of recommendations, allowing users to review past performance and refine their strategies.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.
- The interface for accessing recommendations should be intuitive and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards.

## Social trading features for sharing and copying strategies

### Description

The 'Social trading features for sharing and copying strategies' in the 'my-test-juan-0304' app enable users to interact with and learn from each other by sharing their trading strategies and copying successful strategies from other users. This feature fosters a collaborative trading environment, allowing users to follow top-performing traders, analyze their strategies, and replicate them in their own portfolios. It is designed to enhance user engagement, provide educational value, and improve trading outcomes by leveraging collective intelligence within the community.

### Acceptance Criteria

- Users must be able to share their trading strategies with the community, including detailed parameters, performance metrics, and historical results.
- The app should provide a leaderboard or ranking system to highlight top-performing traders based on metrics such as win rate, profit factor, and drawdown.
- Users must be able to browse and search for shared strategies using filters such as performance metrics, risk level, trading style, and popularity.
- The feature must include a 'Copy Strategy' button that allows users to replicate a selected strategy in their own portfolio with adjustable parameters to suit their preferences.
- Provide detailed analytics and insights for copied strategies, including real-time performance tracking and comparison with the original strategy.
- Include a notification system to alert users when a strategy they are following is updated or achieves significant milestones.
- Ensure privacy controls for users who share strategies, allowing them to choose whether their strategies are public, private, or shared with specific individuals.
- Implement a feedback and rating system for shared strategies, enabling users to leave reviews and comments to help others assess the quality and reliability of the strategy.
- The feature must support integration with the app's risk management tools, ensuring copied strategies adhere to user-defined risk thresholds and settings.
- Provide educational resources and tutorials to help users understand how to evaluate and copy strategies effectively.
- Ensure compatibility with the app's subscription tiers, with advanced social trading features available to premium users while offering basic functionality to free-tier users.
- Include error handling to manage potential issues such as strategy incompatibility, insufficient funds, or technical failures during strategy replication.
- The feature must comply with relevant regulations and guidelines, ensuring transparency and ethical practices in social trading.
- Test the feature across different devices and operating systems to ensure consistent functionality and user experience.

## Customizable strategy templates for advanced users

### Description

The 'Customizable strategy templates for advanced users' feature in the 'my-test-juan-0304' app empowers experienced traders and fund managers to create, modify, and save personalized trading strategies tailored to their specific needs. This feature provides a flexible interface for adjusting parameters such as entry and exit conditions, risk management settings, and technical indicators. Users can leverage pre-built templates as a starting point or design strategies from scratch, enabling them to optimize their trading performance and adapt to dynamic market conditions. The feature is designed to enhance user autonomy and creativity while maintaining robust functionality and ease of use.

## Acceptance Criteria

- The feature must provide a library of pre-built strategy templates, including popular trading strategies such as scalping, swing trading, and trend following.
- Users must be able to customize templates by adjusting parameters such as entry and exit conditions, stop-loss and take-profit settings, position sizing, and technical indicators.
- The customization interface should be intuitive, featuring drag-and-drop functionality, sliders, dropdown menus, and real-time previews of strategy performance.
- Users must be able to save their customized strategies for future use, with options to name and categorize them for easy access.
- The app should allow users to test their customized strategies using historical data through the backtesting engine, providing detailed performance metrics such as win rate, profit factor, and drawdown.
- Users must be able to simulate their strategies in a paper trading mode with real-time market data to validate their effectiveness before live trading.
- The feature should support exporting and importing strategy templates, enabling users to share their strategies with others or transfer them across accounts.
- Customizable strategy templates must integrate seamlessly with other app features, such as AI-driven insights, risk management tools, and analytics dashboards.
- The feature must include error handling to prevent invalid configurations and provide clear feedback to users about any issues with their strategy setup.
- Ensure compatibility with all supported devices and operating systems, including desktop and mobile platforms, with a responsive design for consistent user experience.
- The feature must adhere to security best practices, ensuring that user-created strategies are stored securely and are not accessible to unauthorized users.
- Provide interactive tutorials and documentation to guide users through the process of customizing and optimizing their strategies.
- Include analytics to track user engagement with the feature, such as the number of strategies created, modified, and tested, to inform future improvements.

# Risk Management

## Maximum trade size controls (% of portfolio)

### Description

The 'Maximum trade size controls (% of portfolio)' feature in the 'my-test-juan-0304' app allows users to set limits on the maximum percentage of their portfolio that can be allocated to a single trade. This feature is designed to help users manage risk effectively by preventing overexposure to individual trades. It provides a customizable interface where users can define their preferred trade size limits based on their risk tolerance and trading strategy. By enforcing these limits, the feature ensures disciplined trading practices and safeguards against significant losses due to high-risk trades.

### Acceptance Criteria

- Users must be able to access the 'Maximum trade size controls' feature from the risk management settings section of the app.
- The feature must allow users to set a maximum trade size as a percentage of their total portfolio value, with a default value of 5% for new users.
- The app should provide visual indicators (e.g., sliders, input fields) for users to easily adjust the trade size limit.
- The system must enforce the trade size limit during trade execution, preventing users from placing trades that exceed the defined percentage of their portfolio.
- Users should receive a warning notification if they attempt to place a trade that exceeds the set limit, along with an option to adjust their settings or cancel the trade.
- The feature must support different portfolio types, including individual accounts and fund manager accounts managing multiple portfolios.
- The app should provide analytics and insights on trade size limits, including historical data on trades that adhered to or exceeded the set limits.
- The feature must be compatible with all trading strategies implemented in the app, including scalping, swing trading, and trend following.
- Ensure the feature is tested across different devices and operating systems for consistent functionality and user experience.
- The interface must adhere to accessibility standards, ensuring usability for all users, including those with disabilities.
- Include error handling to manage scenarios where portfolio data is unavailable or outdated, providing users with clear feedback and resolution options.
- Integrate analytics to track user behavior and preferences related to trade size limits, enabling future improvements to the feature.

## Stop-loss settings (fixed and trailing)

### Description

The 'Stop-loss settings (fixed and trailing)' feature in the 'my-test-juan-0304' app provides users with the ability to manage risk by automatically limiting potential losses on trades. Fixed stop-loss settings allow users to define a specific price level at which a trade will be exited, ensuring losses are capped at a predetermined amount. Trailing stop-loss settings dynamically adjust the stop-loss level as the trade moves in a favorable direction, locking in profits while still protecting against downside risk. This feature is essential for maintaining disciplined trading practices and mitigating the impact of market volatility.

### Acceptance Criteria

- Users must be able to set fixed stop-loss levels by specifying a price or percentage below the entry price for each trade.
- Users must be able to enable trailing stop-loss settings, which automatically adjust the stop-loss level as the trade moves in a favorable direction, based on a user-defined percentage or price increment.
- The app must provide a clear and intuitive interface for configuring stop-loss settings, including visual indicators on charts to display the stop-loss levels.
- Stop-loss settings must be applicable to individual trades as well as portfolio-wide configurations, allowing users to manage risk at both granular and macro levels.
- The app must send real-time notifications (via in-app alerts, email, or SMS) when a stop-loss is triggered, ensuring users are informed of trade exits immediately.
- The stop-loss mechanism must execute trades promptly and reliably when the stop-loss level is reached, minimizing slippage and ensuring compliance with user-defined risk parameters.
- Users must be able to review and modify stop-loss settings for active trades at any time, with changes taking effect immediately.
- The app must provide analytics and reporting on stop-loss performance, including metrics such as the number of trades exited via stop-loss and the impact on overall portfolio performance.
- Stop-loss settings must integrate seamlessly with other risk management features, such as maximum trade size controls and volatility-adjusted position sizing, to provide a comprehensive risk management solution.
- The feature must be compatible with all supported exchanges and trading pairs, ensuring consistent functionality across different markets.
- The app must include error handling to manage potential issues, such as exchange-specific limitations or network connectivity problems, ensuring stop-loss settings remain reliable and effective.

- The stop-loss feature must adhere to industry best practices for security and compliance, ensuring user data and trade information are protected at all times.

## Take-profit settings

### Description

The 'Take-profit settings' feature in the 'my-test-juan-0304' app allows users to define specific price levels or percentage gains at which their trades will automatically close to secure profits. This feature is essential for effective risk management and helps users lock in gains without needing to constantly monitor the market. It provides flexibility for users to set fixed take-profit levels or dynamic thresholds based on market conditions, ensuring alignment with their trading strategies and goals.

### Acceptance Criteria

- Users must be able to set take-profit levels as either a fixed price or a percentage gain relative to the entry price.
- The feature must support dynamic take-profit settings, such as trailing take-profit, which adjusts the threshold based on market movements to maximize gains.
- Take-profit settings must be accessible and configurable within the trade setup interface for all supported trading pairs and strategies.
- The app must provide visual indicators on charts to display active take-profit levels for ongoing trades, ensuring clarity for users.
- Notifications must be sent to users via in-app alerts, email, or SMS when a take-profit level is triggered and the trade is closed.
- The feature must integrate seamlessly with other risk management tools, such as stop-loss settings, to provide comprehensive trade control.
- Take-profit settings must be compatible with all integrated exchanges and trading pairs, ensuring consistent functionality across platforms.
- The app must ensure high performance and reliability in executing take-profit orders, even during periods of high market volatility.
- Error handling mechanisms must be implemented to manage potential issues, such as exchange API failures or network disruptions, with appropriate user feedback.
- The feature must adhere to user privacy and security standards, ensuring that sensitive trade data is encrypted and protected.
- Interactive tutorials and tooltips must be available to guide users in configuring take-profit settings effectively.
- Analytics must be integrated to track the performance of take-profit settings, providing insights into their effectiveness and areas for improvement.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and user experience.
- Accessibility standards must be met, ensuring that users with disabilities can configure and utilize take-profit settings without barriers.

## Maximum daily trading volume limits

### Description

The 'Maximum daily trading volume limits' feature in the 'my-test-juan-0304' app is designed to help users manage their trading activity by setting predefined limits on the total trading volume they can execute within a single day. This feature is particularly useful for mitigating risks associated with overtrading and ensuring adherence to personal or organizational trading policies. Users can customize these limits based on their portfolio size, risk tolerance, or compliance requirements. The feature provides real-time monitoring and alerts to notify users when they approach or exceed their daily trading volume limit, ensuring proactive risk management.

### Acceptance Criteria

- Users must be able to set a maximum daily trading volume limit in the app, either as a percentage of their portfolio or a fixed monetary value.
- The feature must provide real-time tracking of the user's trading volume throughout the day, updating dynamically as trades are executed.
- Notifications must be sent to users when they approach 80% of their daily trading volume limit, and a final alert must be sent when the limit is reached.
- Once the daily trading volume limit is exceeded, the app must automatically restrict further trade execution for the remainder of the day, unless the user manually overrides the limit with appropriate justification (e.g., compliance approval).
- The feature must include a dashboard or visualization tool that displays the user's current trading volume, remaining limit, and historical trading volume trends for better decision-making.
- The app must allow users to customize the limit settings based on their trading preferences, risk tolerance, or compliance requirements.
- Fund Managers must be able to set trading volume limits for individual accounts they manage, ensuring compliance across multiple portfolios.
- The feature must include error handling to ensure accurate tracking of trading volume, even in cases of network disruptions or exchange-specific issues.
- The feature must integrate seamlessly with the app's risk management dashboard, providing a holistic view of the user's exposure and risk metrics.
- Ensure compatibility with all supported exchanges and trading pairs, accurately aggregating trading volume data across multiple platforms.



- The feature must adhere to relevant financial regulations and compliance standards, ensuring that trading limits are enforced in a secure and transparent manner.
- Include analytics to track user behavior and identify patterns in trading volume limit usage, which can be used to refine the feature and provide personalized recommendations.

## Maximum drawdown thresholds

### Description

The 'Maximum drawdown thresholds' feature in the 'my-test-juan-0304' app is designed to help users manage risk by setting predefined limits on the maximum allowable portfolio drawdown. This feature ensures that users can safeguard their investments by automatically pausing trading activities or triggering alerts when the portfolio's value drops below a specified percentage or amount. By providing a proactive mechanism to mitigate losses, this feature enhances risk management and promotes disciplined trading practices.

### Acceptance Criteria

- Users must be able to set maximum drawdown thresholds as a percentage of their portfolio value or as a fixed monetary amount.
- The feature must provide an intuitive interface for configuring drawdown thresholds, including sliders, input fields, and visual aids such as charts showing historical drawdown data.
- When the portfolio value reaches the specified drawdown threshold, the app must automatically pause all trading activities and notify the user via in-app notifications, email, and SMS (if enabled).
- The app must provide users with the option to customize the response to reaching the drawdown threshold, such as pausing trading, liquidating positions, or sending alerts only.
- The feature must include a dashboard or report that displays real-time drawdown metrics, historical drawdown data, and the current status of the portfolio relative to the threshold.
- Users must be able to adjust or disable the drawdown threshold settings at any time, with changes taking effect immediately.
- The app must include error handling to ensure that drawdown calculations are accurate and unaffected by issues such as delayed market data or API failures.
- The feature must support multi-account setups, allowing Fund Managers to set individual drawdown thresholds for each account they manage.
- The app must provide educational resources or tooltips explaining the concept of drawdown and how to effectively use the feature for risk management.
- The feature must be compatible with all trading strategies implemented in the app, ensuring seamless integration with other risk management tools such as stop-loss and take-profit settings.
- The drawdown threshold functionality must demonstrate high performance and reliability, with real-time monitoring and immediate response to threshold breaches.
- The feature must adhere to security and compliance standards, ensuring that sensitive portfolio data is encrypted and protected from unauthorized access.
- The interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Risk visualization dashboard for current exposure

### Description

The 'Risk visualization dashboard for current exposure' feature in the 'my-test-juan-0304' app provides users with a comprehensive and interactive dashboard to monitor their portfolio's risk exposure in real-time. This feature enables users to visualize key risk metrics, such as portfolio allocation, exposure to specific assets or sectors, and potential drawdowns. The dashboard is designed to help users make informed decisions by presenting data in an intuitive and actionable format, including charts, graphs, and alerts. It is essential for maintaining effective risk management practices and ensuring users can proactively address potential vulnerabilities in their trading strategies.

### Acceptance Criteria

- The dashboard must be accessible from the main navigation menu and prominently displayed within the 'Risk Management' section of the app.
- Users must be able to view real-time data on portfolio allocation, asset exposure, and risk metrics, including maximum drawdown and volatility levels.
- The dashboard should include visual elements such as pie charts, bar graphs, and heat maps to represent risk exposure clearly and intuitively.
- Users must be able to filter and customize the dashboard view based on specific criteria, such as asset class, sector, or trading strategy.
- The dashboard should provide alerts and notifications for significant risk events, such as exceeding maximum drawdown thresholds or high volatility periods.
- Users must be able to drill down into specific assets or sectors to view detailed risk metrics and historical performance data.
- The dashboard must support multi-account views for Fund Managers, allowing them to monitor risk exposure across multiple portfolios simultaneously.
- The feature should integrate with AI-driven insights to provide recommendations for mitigating risk and optimizing portfolio allocation.

- The dashboard must demonstrate high performance, with real-time updates and minimal latency, even during periods of high market activity.
- Ensure compatibility with both desktop and mobile versions of the app, with responsive design for seamless user experience across devices.
- The dashboard must adhere to accessibility standards (WCAG) to ensure usability for all users, including those with disabilities.
- Include error handling to manage potential issues with data retrieval or visualization, providing users with clear feedback and alternative options.
- Integrate analytics to track user interactions with the dashboard, enabling continuous improvement based on user behavior and feedback.

## **Risk alert system with notifications via in-app, email, and SMS**

### **Description**

The 'Risk alert system with notifications via in-app, email, and SMS' feature in the 'my-test-juan-0304' app is designed to proactively notify users about critical risk-related events in their trading activities. This feature ensures that users are promptly informed of potential issues such as approaching stop-loss thresholds, exceeding maximum trade size limits, or encountering extreme market volatility. Notifications are delivered through multiple channels, including in-app alerts, email, and SMS, providing flexibility and ensuring users stay updated regardless of their preferred communication method. By offering timely and actionable insights, this feature helps users mitigate risks effectively and maintain control over their trading strategies.

### **Acceptance Criteria**

- The risk alert system must be configurable by users to define specific risk thresholds, such as stop-loss levels, maximum trade size, and drawdown limits.
- Notifications must be delivered promptly through in-app alerts, email, and SMS, ensuring users are informed in real-time about critical risk events.
- Users must be able to customize notification preferences, including the choice of delivery channels (in-app, email, SMS) and the frequency of alerts.
- The system should provide detailed information in each notification, including the type of risk event, affected trades or portfolios, and recommended actions to mitigate the risk.
- In-app alerts must be prominently displayed on the dashboard or trading interface, ensuring visibility without disrupting the user experience.
- Email notifications must include a clear subject line and concise content, with links to relevant sections of the app for further action.
- SMS notifications must be concise and formatted for quick readability, providing essential details about the risk event and a call-to-action if necessary.
- The risk alert system must support multi-language notifications to cater to users from different regions and language preferences.
- The system must demonstrate high reliability and uptime, ensuring notifications are delivered without delays or failures, even during high market activity periods.
- Include error handling mechanisms to manage issues such as failed email or SMS delivery, with fallback options to retry or notify users through alternate channels.
- The system must comply with data privacy regulations, ensuring user contact information is securely stored and used only for notification purposes.
- Integrate analytics to track user engagement with notifications, such as open rates and response actions, to refine the system and improve its effectiveness.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.
- The notification interface must adhere to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## **Circuit breaker mechanism for automatic trading pause during extreme volatility**

### **Description**

The 'Circuit breaker mechanism for automatic trading pause during extreme volatility' feature in 'my-test-juan-0304' app is designed to protect users from significant losses during periods of extreme market fluctuations. This feature automatically halts trading activities when predefined volatility thresholds are breached, ensuring that users' portfolios are safeguarded from adverse market conditions. The mechanism monitors real-time market data and triggers the pause based on volatility metrics such as price swings, trading volume spikes, or other indicators. Users are notified immediately when the circuit breaker is activated, and trading resumes only after conditions stabilize or upon manual override. This feature is essential for maintaining risk control and providing users with peace of mind during unpredictable market scenarios.

### **Acceptance Criteria**

- The circuit breaker mechanism must continuously monitor real-time market data for volatility indicators, including price swings, trading volume spikes, and other relevant metrics.

- Users must be able to configure the volatility thresholds that trigger the circuit breaker, allowing customization based on individual risk tolerance.
- When the circuit breaker is activated, all trading activities (including automated trades and manual orders) must be paused immediately to prevent further exposure to market risks.
- The app must provide clear and immediate notifications to users when the circuit breaker is triggered, including details about the reason for activation and the affected trades.
- Users must have the option to manually override the circuit breaker and resume trading, with a confirmation prompt to ensure intentional action.
- Trading activities must automatically resume once market conditions stabilize and fall below the predefined volatility thresholds, or upon user intervention.
- The circuit breaker mechanism must be integrated with the risk visualization dashboard, allowing users to view real-time updates on market conditions and circuit breaker status.
- The feature must demonstrate high performance and reliability, ensuring that the circuit breaker is triggered promptly during extreme volatility without delays or errors.
- Include error handling to manage potential failures in market data retrieval or system malfunctions, ensuring the app remains stable and provides feedback to the user about the issue.
- The circuit breaker mechanism must comply with relevant financial regulations and industry standards for risk management in trading platforms.
- Ensure compatibility with all supported exchanges and trading pairs, accounting for exchange-specific volatility metrics and limitations.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and user experience.
- Provide detailed documentation and tutorials to help users understand how the circuit breaker mechanism works and how to configure it effectively.

## Volatility-adjusted position sizing

### Description

The 'Volatility-adjusted position sizing' feature in the 'my-test-juan-0304' app enables users to dynamically adjust their trade sizes based on market volatility. By leveraging real-time volatility metrics, this feature ensures that users can optimize their risk exposure while maintaining consistent portfolio management practices. It helps traders and investors mitigate risks during periods of high market turbulence and capitalize on opportunities during stable conditions. This feature is essential for enhancing risk management strategies and improving overall trading performance.

### Acceptance Criteria

- The feature must calculate position sizes based on real-time market volatility metrics, such as Average True Range (ATR) or standard deviation of price movements.
- Users must be able to enable or disable volatility-adjusted position sizing from their account settings or trading strategy configuration interface.
- The app should provide a clear visualization of how volatility impacts position sizing, including charts or graphs showing historical volatility trends and their effect on trade sizes.
- Position sizing adjustments must be applied automatically during trade execution, ensuring seamless integration with the user's trading strategies.
- The feature must allow users to set custom thresholds for volatility levels, enabling them to define their own risk tolerance and position sizing rules.
- Volatility data must be sourced from reliable market data providers and updated in real-time to ensure accuracy and relevance.
- The app should include a simulation mode where users can test the impact of volatility-adjusted position sizing on historical trades and market conditions.
- Provide detailed analytics and reporting on the performance of trades executed with volatility-adjusted position sizing, including metrics like risk-adjusted returns and drawdown reduction.
- Ensure compatibility with all trading strategies supported by the app, including scalping, swing trading, and trend following.
- Include error handling mechanisms to manage scenarios where volatility data is unavailable or delayed, ensuring the feature remains stable and provides fallback options for users.
- The feature must adhere to user privacy and data security standards, ensuring that sensitive trading data is protected during calculations and reporting.
- Ensure the feature is accessible and intuitive, with clear instructions and tooltips for users unfamiliar with volatility-adjusted position sizing.
- Test the feature across different devices and operating systems to ensure consistent functionality and performance.

## AI-driven risk mitigation strategies

### Description

The 'AI-driven risk mitigation strategies' feature in 'my-test-juan-0304' app leverages advanced artificial intelligence algorithms to proactively identify and mitigate potential risks in trading activities. This feature analyzes real-time market data, historical trends, and user-specific trading patterns to provide actionable insights and automated adjustments to trading strategies. By dynamically adapting to market

conditions, the feature helps users minimize losses, optimize risk exposure, and maintain portfolio stability. It is designed to cater to the diverse needs of Active Traders, Passive Investors, and Fund Managers, ensuring a tailored approach to risk management.

## Acceptance Criteria

- The AI system must continuously monitor real-time market data and user trading activity to identify potential risks, such as high volatility, sudden price drops, or unfavorable market conditions.
- The feature must provide users with actionable recommendations for mitigating risks, such as adjusting position sizes, setting tighter stop-loss limits, or temporarily pausing trading activities.
- The AI must be capable of automatically implementing risk mitigation measures based on predefined user preferences, such as activating circuit breakers or reducing exposure to volatile assets.
- Risk mitigation strategies must be customizable, allowing users to define their risk tolerance levels, preferred trading parameters, and specific conditions for AI intervention.
- The feature must include a dashboard that visually represents current risk exposure, AI-driven adjustments, and the impact of mitigation strategies on portfolio performance.
- The AI algorithms must be tested and validated to ensure accuracy and reliability in identifying risks and recommending appropriate actions.
- The feature must support multi-asset portfolios, providing risk mitigation strategies for various asset classes, including cryptocurrencies, stocks, and derivatives.
- Users must receive real-time notifications via in-app alerts, email, or SMS when the AI detects significant risks or implements mitigation measures.
- The feature must include detailed logs and reports of AI-driven actions, allowing users to review and analyze the effectiveness of the risk mitigation strategies over time.
- The AI system must prioritize user privacy and data security, ensuring that sensitive trading information is protected and not exposed during risk analysis or mitigation processes.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems to ensure consistent functionality.
- The risk mitigation strategies must adhere to relevant financial regulations and compliance standards, ensuring ethical and legal trading practices.
- The feature must integrate seamlessly with other app functionalities, such as trading strategies, analytics, and reporting, to provide a cohesive user experience.

# Analytics & Reporting

## Comprehensive dashboards for trading performance insights

### Description

The 'Comprehensive dashboards for trading performance insights' feature in the 'my-test-juan-0304' app provides users with an advanced and visually appealing interface to analyze their trading performance. This feature enables users to monitor key metrics such as profit and loss, win rate, drawdown, and portfolio growth over time. The dashboards are customizable, allowing users to focus on the data most relevant to their trading strategies and goals. With real-time updates and historical data integration, users can make informed decisions to optimize their trading performance. The feature is designed to cater to the needs of all user roles, from active traders seeking detailed analytics to fund managers requiring aggregated insights across multiple accounts.

### Acceptance Criteria

- The dashboard must be accessible from the main navigation menu and prominently displayed for easy access.
- Users must be able to view key performance metrics, including profit and loss, win rate, drawdown, portfolio growth, and trade frequency.
- The dashboard should support real-time updates, ensuring users have access to the latest trading data without delays.
- Historical data integration must be implemented, allowing users to analyze performance trends over customizable timeframes (e.g., daily, weekly, monthly, yearly).
- The dashboard must be customizable, enabling users to add, remove, or rearrange widgets based on their preferences and trading goals.
- Interactive charts and graphs should be included to visually represent trading performance metrics, with options for zooming and filtering data.
- Users must be able to export dashboard data in multiple formats (e.g., CSV, PDF) for offline analysis or reporting purposes.
- The dashboard should include benchmarking capabilities, allowing users to compare their performance against market indices or other traders.
- Fund managers must have access to aggregated insights across multiple accounts, with the ability to drill down into individual account performance.
- The dashboard must be optimized for both desktop and mobile platforms, ensuring seamless access across devices.
- Accessibility features must be implemented, adhering to WCAG standards to ensure usability for all users, including those with disabilities.
- Error handling must be included to manage potential issues with data retrieval or visualization, providing clear feedback to users in case of problems.
- The dashboard must demonstrate high performance and responsiveness, even when handling large datasets or complex visualizations.
- Integrate analytics tracking to monitor user interactions with the dashboard, providing insights for future improvements and feature enhancements.

## Customizable report builder for trading metrics

### Description

The 'Customizable report builder for trading metrics' feature in the 'my-test-juan-0304' app empowers users to create tailored reports that provide insights into their trading performance and metrics. This feature allows users to select specific data points, visualizations, and formats to suit their individual needs. It supports advanced filtering, sorting, and grouping options, enabling users to analyze their trading activity in depth. The customizable report builder is designed to enhance decision-making by providing actionable insights and facilitating compliance with financial reporting requirements.

### Acceptance Criteria

- Users must be able to access the report builder from the Analytics & Reporting section of the app.
- The report builder must allow users to select specific trading metrics, such as win rate, profit factor, drawdown, and trade volume, for inclusion in the report.
- Users must be able to apply filters (e.g., date range, trading strategy, asset type) to refine the data displayed in the report.
- The report builder must support multiple visualization options, including tables, charts, graphs, and heat maps, to cater to diverse user preferences.
- Users must be able to group and sort data within the report based on selected criteria, such as trading strategy or asset class.
- Reports must be exportable in multiple formats, including PDF, Excel, and CSV, to facilitate sharing and external analysis.
- The feature must include pre-built templates for common reporting needs, such as performance benchmarking and compliance reporting, with options for customization.
- Users must be able to save and reuse custom report configurations for future use, improving efficiency and consistency.
- The report builder must include a preview mode, allowing users to review the report layout and content before finalizing it.
- Reports must be optimized for both desktop and mobile viewing, ensuring accessibility across devices.
- The feature must include error handling to manage issues such as missing data or invalid filters, providing clear feedback to users.

- The report builder must adhere to data security and privacy standards, ensuring sensitive trading information is protected during report generation and export.
- The feature must support role-based access control, ensuring that only authorized users can access and generate reports based on their permissions.
- Include analytics to track user interactions with the report builder, providing insights into popular metrics and configurations for future improvements.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.

## Performance benchmarking against market indices and manual trading

### Description

The 'Performance benchmarking against market indices and manual trading' feature in the 'my-test-juan-0304' app enables users to evaluate their trading performance by comparing it against key market indices (e.g., S&P 500, NASDAQ) and manual trading results. This feature provides detailed insights into how automated strategies and trades executed through the app measure up against broader market trends and traditional trading methods. By offering clear metrics and visualizations, users can identify areas for improvement, validate the effectiveness of their strategies, and make informed decisions to optimize their trading approach.

### Acceptance Criteria

- The feature must allow users to select specific market indices (e.g., S&P 500, NASDAQ, Dow Jones) for benchmarking their trading performance.
- Users must be able to compare their automated trading results against manual trading outcomes, with detailed metrics such as ROI, win rate, profit factor, and drawdown.
- Provide visualizations such as line charts, bar graphs, and heat maps to illustrate performance comparisons over customizable timeframes (e.g., daily, weekly, monthly, yearly).
- Include a summary dashboard that highlights key performance metrics, such as percentage outperformance or underperformance relative to selected benchmarks.
- Enable users to filter benchmarking results by trading strategy, asset class, or trading pair for granular analysis.
- The app must support historical data analysis, allowing users to benchmark past performance against market indices and manual trading results.
- Integrate predictive analytics to forecast potential future performance based on historical trends and current market conditions.
- Ensure benchmarking data is updated in real-time, reflecting the latest market movements and trading activity.
- Provide actionable insights and recommendations based on benchmarking results, such as adjustments to trading strategies or portfolio allocations.
- Include export functionality for benchmarking reports, allowing users to download or share their performance data in formats such as PDF or CSV.
- Ensure compatibility with the app's existing analytics dashboard, allowing seamless navigation between benchmarking and other reporting features.
- The benchmarking feature must adhere to data privacy and security standards, ensuring sensitive user data is protected during analysis and reporting.
- Test the feature across different devices and operating systems to ensure consistent functionality and user experience.
- Provide an interactive tutorial or help section to guide users on how to use the benchmarking feature effectively.

## Predictive analytics for trading success and market trends

### Description

The 'Predictive analytics for trading success and market trends' feature in the 'my-test-juan-0304' app leverages advanced AI and machine learning algorithms to analyze historical and real-time market data, providing users with actionable insights into potential trading opportunities and market movements. This feature empowers users to make informed decisions by forecasting trade success probabilities, identifying emerging trends, and highlighting risk factors. It is designed to cater to the diverse needs of Active Traders, Passive Investors, and Fund Managers, offering tailored predictions based on their trading strategies and preferences.

### Acceptance Criteria

- The predictive analytics feature must be accessible from the main analytics dashboard, with a dedicated section for market trend forecasts and trade success probabilities.
- Users must be able to view predictions for specific trading pairs, asset classes, or market sectors, with customizable filters for timeframes (e.g., intraday, weekly, monthly).
- The app should utilize AI-driven models to provide probability scores for trade success, along with visualizations such as graphs, charts, and heat maps to represent market trends and predictions.
- Predictions must be updated in real-time, incorporating the latest market data and volatility metrics to ensure accuracy and relevance.
- The feature should include a confidence level indicator for each prediction, helping users assess the reliability of the insights provided.

- Users must be able to access historical prediction accuracy metrics to evaluate the performance of the predictive analytics engine over time.
- The app should allow users to set alerts for specific predictive thresholds (e.g., high probability of success or significant market trend changes), with notifications delivered via in-app, email, or SMS.
- The predictive analytics engine must be optimized for high performance, ensuring quick response times even during periods of high market activity.
- The feature should include error handling mechanisms to manage potential issues with data retrieval or algorithm processing, providing users with clear feedback in case of disruptions.
- The predictive analytics must adhere to data privacy and security standards, ensuring that sensitive user information and trading data are protected.
- The feature should be compatible with all supported devices and operating systems, ensuring a consistent user experience across platforms.
- The interface must be designed with accessibility in mind, adhering to WCAG (Web Content Accessibility Guidelines) standards to ensure usability for all users.
- Integrate analytics tracking to monitor user engagement with the predictive analytics feature, providing insights for future improvements and refinements.

## Real-time analytics dashboard for trade execution and performance

### Description

The 'Real-time analytics dashboard for trade execution and performance' feature in 'my-test-juan-0304' app provides users with a dynamic and interactive dashboard that delivers live insights into their trading activities and performance metrics. This feature is designed to empower users by offering actionable data, enabling them to make informed decisions in real-time. The dashboard includes visualizations such as charts, graphs, and tables that display key metrics like trade success rates, profit and loss, execution speed, and market trends. It is tailored to meet the needs of different user roles, ensuring that Active Traders, Passive Investors, and Fund Managers can monitor and optimize their trading strategies effectively.

### Acceptance Criteria

- The dashboard must be accessible from the main navigation menu and prominently displayed for easy access.
- Users must be able to view real-time updates on trade execution metrics, including execution speed, slippage, and order fill rates.
- Performance metrics such as profit and loss, win rate, drawdown, and return on investment (ROI) must be displayed in an intuitive and visually appealing format (e.g., charts, graphs, tables).
- The dashboard should include filters and sorting options to allow users to customize the view based on specific timeframes, trading pairs, or strategies.
- Data visualizations must be interactive, allowing users to hover over elements for detailed information and drill down into specific metrics or trades.
- The dashboard must provide alerts and notifications for significant changes in performance metrics or market conditions, ensuring users can respond promptly.
- Users should be able to export analytics data in multiple formats (e.g., CSV, PDF) for offline analysis or reporting purposes.
- The dashboard must support multi-device compatibility, ensuring seamless access on desktop, mobile, and tablet devices.
- Ensure high performance and responsiveness of the dashboard, even when handling large datasets or high-frequency updates.
- Include error handling mechanisms to manage potential issues with data retrieval or visualization, providing clear feedback to users in case of errors.
- The dashboard must adhere to data privacy and security standards, ensuring sensitive trading information is protected and only accessible to authorized users.
- Integrate AI-driven insights to provide recommendations for improving trading performance based on historical and real-time data.
- The dashboard must be tested across different user roles to ensure it meets the specific needs of Active Traders, Passive Investors, and Fund Managers.
- Ensure accessibility compliance with WCAG standards, making the dashboard usable for individuals with disabilities.

## Historical trade analysis with detailed metrics (win rate, profit factor, drawdown)

### Description

The 'Historical trade analysis with detailed metrics' feature in the 'my-test-juan-0304' app provides users with comprehensive insights into their past trading performance. By analyzing historical trades, users can view key metrics such as win rate, profit factor, and drawdown, enabling them to evaluate the effectiveness of their trading strategies and identify areas for improvement. This feature is designed to support informed decision-making and enhance trading outcomes by offering actionable data in an intuitive and visually appealing format.

### Acceptance Criteria

- The feature must be accessible from the analytics dashboard and clearly labeled for easy navigation.
- Users must be able to view historical trade data organized by timeframes (e.g., daily, weekly, monthly, yearly).
- The app should calculate and display key metrics, including win rate (percentage of successful trades), profit factor (ratio of gross profit to gross loss), and drawdown (maximum loss from peak to trough).
- Visualizations such as charts and graphs must be included to represent trends in win rate, profit factor, and drawdown over time.

- Users must be able to filter historical trade data by trading strategy, asset type, and exchange to focus on specific subsets of their performance.
- The feature should provide comparative analysis, allowing users to benchmark their performance against market indices or other traders within the app (if social trading is enabled).
- Export functionality must be available, enabling users to download historical trade analysis reports in formats such as PDF or CSV for external review or compliance purposes.
- The app must ensure high performance and responsiveness when processing large datasets of historical trades, with minimal delays.
- Error handling must be implemented to manage scenarios where historical data is incomplete or unavailable, providing users with clear feedback and suggestions for resolution.
- The feature must adhere to data privacy and security standards, ensuring that sensitive trading information is protected and accessible only to authorized users.
- The interface must be designed to be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards to accommodate users with disabilities.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.

## Heat maps for market activity and volatility

### Description

The 'Heat maps for market activity and volatility' feature in the 'my-test-juan-0304' app provides users with a visual representation of market activity and volatility across various trading pairs, sectors, or asset classes. This feature enables users to quickly identify areas of high or low activity, as well as regions of significant price fluctuations, helping them make informed trading decisions. The heat maps are color-coded and interactive, offering a clear and intuitive way to analyze market trends and volatility patterns in real-time or over historical periods.

### Acceptance Criteria

- The heat map must be accessible from the 'Analytics & Reporting' section of the app and should be prominently displayed for easy navigation.
- Users must be able to view heat maps for different asset classes, trading pairs, or sectors, with the ability to filter by specific criteria such as market capitalization, volume, or volatility levels.
- The heat map should use a color-coded system to represent varying levels of market activity and volatility, with a clear legend explaining the color scheme.
- Users must be able to toggle between real-time data and historical data for the heat maps, with options to select specific timeframes (e.g., last 24 hours, 7 days, 30 days).
- Interactive features must allow users to hover over or click on specific areas of the heat map to view detailed metrics, such as trading volume, price change percentage, and volatility index.
- The heat map should support zooming and panning for a more detailed view of specific regions or assets.
- Users must be able to export heat map data as an image or CSV file for further analysis or reporting purposes.
- The feature must include a search functionality to quickly locate specific assets or trading pairs within the heat map.
- The heat map must update in real-time for live market data, ensuring users have access to the most current information.
- The feature must be optimized for both desktop and mobile platforms, ensuring a seamless user experience across devices.
- The heat map must adhere to accessibility standards, including support for colorblind users (e.g., alternative color schemes or patterns).
- The feature must demonstrate high performance, with minimal loading times even when displaying large datasets.
- Include error handling to manage potential issues with data retrieval or rendering, providing users with clear feedback in case of errors.
- Integrate analytics to track user interactions with the heat map, such as the most viewed assets or sectors, to inform future feature improvements.

## Integration with tax reporting tools for financial compliance

### Description

The 'Integration with tax reporting tools for financial compliance' feature in 'my-test-juan-0304' app streamlines the process of generating tax-related reports for users engaged in trading activities. This feature enables seamless integration with popular tax reporting tools, ensuring accurate calculation and reporting of trading profits, losses, and other financial metrics required for tax compliance. It simplifies the preparation of tax documents by automatically categorizing transactions, applying relevant tax rules, and generating exportable reports in formats compatible with tax software. This feature is essential for reducing the administrative burden on users and ensuring compliance with financial regulations.

### Acceptance Criteria

- The app must support integration with at least three popular tax reporting tools (e.g., TurboTax, TaxAct, CoinTracker) to cater to diverse user preferences.
- Users must be able to export their trading data in formats compatible with tax reporting tools, such as CSV, XLSX, or JSON.
- The feature should automatically categorize transactions (e.g., buy, sell, fees) and calculate relevant metrics such as capital gains, losses, and taxable income based on user-defined tax settings.



- Provide users with the ability to customize tax settings, including country-specific tax rules, filing status, and applicable deductions.
- Include a dashboard for users to review and verify their tax-related data before exporting or submitting it to tax reporting tools.
- Ensure the integration adheres to data privacy and security standards, encrypting sensitive financial information during transmission and storage.
- The app must provide error handling and validation to ensure the accuracy of exported data, with clear feedback for users in case of discrepancies or missing information.
- Offer in-app tutorials or guides to help users understand how to use the tax reporting integration effectively.
- Support multi-currency transactions and automatically convert them to the user's preferred reporting currency using real-time exchange rates.
- Include a feature for generating year-end summaries and tax reports, highlighting key metrics such as total gains, losses, and fees paid during the fiscal year.
- Ensure compatibility with the latest tax reporting tools and regularly update the integration to reflect changes in tax laws and regulations.
- Provide analytics to track user engagement with the tax reporting feature, helping improve its functionality and user experience over time.

## Social impact reporting for ethical trading practices

### Description

The 'Social impact reporting for ethical trading practices' feature in 'my-test-juan-0304' app provides users with insights into the ethical and social implications of their trading activities. This feature evaluates the environmental, social, and governance (ESG) impact of investments and trading strategies, helping users align their portfolios with their ethical values. It includes detailed reports on the sustainability and social responsibility of traded assets, enabling users to make informed decisions that contribute positively to society. This feature is designed to promote transparency and accountability in trading practices, fostering a culture of ethical investing.

### Acceptance Criteria

- The feature must provide a dedicated section within the analytics dashboard for social impact reporting, accessible to all applicable user roles.
- Reports must include ESG scores, carbon footprint analysis, and social responsibility metrics for traded assets, sourced from reliable third-party data providers.
- Users must be able to filter and sort reports based on specific criteria, such as industry, asset type, or ESG category.
- The app should offer visualizations, such as charts and graphs, to help users easily interpret the social impact data.
- Reports must include actionable insights, such as recommendations for more sustainable or socially responsible trading options.
- Users should be able to download or export social impact reports in common formats (e.g., PDF, CSV) for external use or compliance purposes.
- The feature must integrate seamlessly with other analytics tools in the app, allowing users to correlate social impact data with financial performance metrics.
- Include a summary section that highlights the overall social impact of the user's portfolio, with comparisons to industry benchmarks or peer groups.
- The feature must support real-time updates to social impact data, ensuring users have access to the latest information for decision-making.
- Ensure compatibility with the latest version of the app and test the feature across different devices and operating systems for consistency.
- The feature must adhere to data privacy and security standards, ensuring that sensitive user information is not exposed in the reports.
- Provide an interactive tutorial or help section to guide users on how to interpret and utilize social impact reports effectively.
- Integrate analytics to track user engagement with the feature, enabling continuous improvement based on user feedback and behavior.

## AI-driven insights for improving trading strategies

### Description

The 'AI-driven insights for improving trading strategies' feature in 'my-test-juan-0304' app leverages advanced artificial intelligence algorithms to analyze trading patterns, market conditions, and historical data. This feature provides actionable recommendations to users for optimizing their trading strategies, enhancing profitability, and minimizing risks. By identifying trends, inefficiencies, and opportunities, the AI-driven insights empower users to make informed decisions tailored to their trading goals and preferences. The feature is designed to cater to all user roles, offering personalized suggestions based on their trading behavior and objectives.

### Acceptance Criteria

- The feature must utilize AI algorithms to analyze historical and real-time trading data, identifying patterns and trends relevant to strategy improvement.
- Recommendations provided by the AI must be tailored to the user's trading role (Active Trader, Passive Investor, Fund Manager) and preferences, ensuring relevance and usability.

- The insights must include actionable suggestions, such as adjustments to strategy parameters, risk management techniques, or market opportunities to explore.
- The feature must display insights in an intuitive and visually appealing format, such as charts, graphs, and summary reports, accessible via the analytics dashboard.
- Users must be able to view detailed explanations of the AI's recommendations, including the rationale behind each suggestion, to build trust and transparency.
- The AI must continuously refine its recommendations based on user feedback, trading outcomes, and evolving market conditions, ensuring adaptability and improvement over time.
- The feature must support integration with other analytics tools in the app, allowing users to cross-reference insights with performance metrics and benchmarks.
- Ensure high performance and quick response times for generating insights, even when processing large datasets or complex trading scenarios.
- Include error handling mechanisms to manage potential issues with data retrieval, AI processing, or system performance, providing clear feedback to users in case of errors.
- The feature must prioritize user privacy and data security, ensuring that sensitive trading data is encrypted and not exposed to unauthorized access.
- Provide accessibility options for users with disabilities, ensuring the insights are presented in formats compatible with assistive technologies.
- Integrate analytics to track user engagement with the feature, such as the frequency of insights usage and the impact on trading performance, for continuous improvement.

# User Experience

## Intuitive step-by-step onboarding process

### Description

The 'Intuitive step-by-step onboarding process' feature in the 'my-test-juan-0304' app is designed to provide a seamless and user-friendly experience for new users. This feature guides users through the onboarding journey in a structured and interactive manner, ensuring they understand the platform's functionalities and can set up their accounts efficiently. It includes tailored steps based on the user's role (Active Trader, Passive Investor, or Fund Manager), offering relevant information and configurations to match their needs. The onboarding process incorporates visual aids, tooltips, and progress tracking to enhance user engagement and reduce friction during account setup.

### Acceptance Criteria

- The onboarding process must begin immediately after a user signs up or logs in for the first time, with an option to skip and revisit later.
- The onboarding flow must be tailored to the user's selected role (Active Trader, Passive Investor, or Fund Manager), presenting relevant features and configurations.
- Each step in the onboarding process must include clear instructions, visual aids, and tooltips to guide users effectively.
- The onboarding process must include a progress tracker, allowing users to see their completion status and navigate back to previous steps if needed.
- Users must be able to customize their preferences during onboarding, such as trading strategies, notification settings, and dashboard layout.
- The onboarding process must introduce key features of the app, including security settings, trading tools, and analytics dashboards, with interactive tutorials where applicable.
- Accessibility features must be integrated into the onboarding process, ensuring usability for users with disabilities (e.g., screen reader compatibility, keyboard navigation).
- The onboarding process must demonstrate high performance, with quick loading times and smooth transitions between steps.
- Error handling must be implemented to manage potential issues during onboarding, such as network connectivity problems or incomplete user inputs, with clear feedback provided to the user.
- The onboarding process must be compatible with both desktop and mobile versions of the app, ensuring consistency across devices.
- Analytics must be integrated to track user behavior during onboarding, providing insights for further optimization and improvement of the process.
- The onboarding process must comply with relevant data privacy regulations, ensuring that user data collected during onboarding is securely stored and handled.

## Simplified dashboard for beginner users with recommended settings

### Description

The 'Simplified dashboard for beginner users with recommended settings' feature in the 'my-test-juan-0304' app is designed to provide an intuitive and streamlined interface for novice users. This dashboard offers pre-configured settings tailored to beginner traders and investors, enabling them to start using the platform effectively without requiring extensive customization or technical knowledge. The feature focuses on reducing complexity while ensuring users have access to essential tools and insights for informed decision-making. By offering a user-friendly experience, this feature aims to enhance onboarding and engagement for new users.

### Acceptance Criteria

- The simplified dashboard must be accessible to users identified as beginners, based on their role or onboarding preferences.
- The dashboard should include pre-configured settings optimized for beginner users, such as default risk management parameters, basic trading strategies, and essential analytics tools.
- The interface must prioritize ease of use, featuring a clean layout with minimal clutter and intuitive navigation options.
- Recommended settings should be based on industry best practices and tailored to the user's selected role (e.g., Passive Investor or Active Trader).
- Users must be able to view key metrics and insights, such as portfolio performance, market trends, and risk exposure, in a simplified format.
- The dashboard should include tooltips, interactive guides, or tutorials to help users understand the functionality of each feature and setting.
- Users must have the option to switch to the advanced dashboard or customize the simplified dashboard settings as they gain experience.
- The feature must be compatible with both desktop and mobile versions of the app, ensuring consistency across devices.
- Performance metrics, such as load times and responsiveness, must meet high standards to ensure a seamless user experience.
- The dashboard must adhere to accessibility standards (e.g., WCAG) to ensure usability for all users, including those with disabilities.
- Include error handling to manage potential issues, such as missing data or connectivity problems, with clear feedback provided to the user.
- Integrate analytics to track user engagement and satisfaction with the simplified dashboard, enabling continuous improvement based on user feedback.

## Advanced configuration tools for experienced traders

### Description

The 'Advanced configuration tools for experienced traders' feature in the 'my-test-juan-0304' app provides a comprehensive suite of customization options tailored for seasoned traders and fund managers. This feature enables users to fine-tune their trading strategies, risk management settings, and portfolio configurations to align with their specific trading goals and market conditions. With an intuitive interface and robust functionality, experienced users can leverage these tools to optimize their trading performance and gain a competitive edge in the market.

### Acceptance Criteria

- The advanced configuration tools must be accessible from the main dashboard or settings menu, ensuring easy navigation for experienced users.
- Users must be able to customize trading parameters such as position sizing, risk thresholds, and strategy-specific settings.
- The feature should include a visual editor for creating and modifying trading strategies, allowing users to adjust parameters like entry/exit conditions, stop-loss levels, and take-profit targets.
- Provide support for multi-strategy configurations, enabling users to run multiple strategies simultaneously and manage them independently.
- Include advanced risk management options, such as volatility-adjusted position sizing and circuit breaker settings, to cater to high-risk trading scenarios.
- Offer integration with analytics tools to provide real-time feedback on the performance of configured settings, helping users refine their strategies.
- Ensure compatibility with AI-driven insights, allowing users to incorporate AI recommendations into their configurations seamlessly.
- Provide a preview mode for users to simulate the impact of their configurations on historical or real-time market data before applying them live.
- Include a version control system for saving, restoring, and comparing different configurations, enabling users to experiment without losing previous setups.
- Ensure the interface is intuitive and user-friendly, with tooltips, guides, and tutorials to assist users in navigating complex configurations.
- The feature must demonstrate high performance and responsiveness, handling large datasets and complex configurations without delays.
- Implement robust error handling to manage invalid configurations or conflicts between settings, providing clear feedback to users.
- Ensure compatibility across devices, including desktop and mobile platforms, with consistent functionality and user experience.
- Adhere to security best practices, ensuring that sensitive configuration data is encrypted and protected from unauthorized access.

## Customizable user interface with drag-and-drop widgets

### Description

The 'Customizable user interface with drag-and-drop widgets' feature in the 'my-test-juan-0304' app empowers users to personalize their dashboard and workspace according to their unique trading and investment needs. This feature allows users to rearrange, add, or remove widgets on their interface, providing flexibility to prioritize the information and tools most relevant to their roles and strategies. By offering a highly adaptable and intuitive interface, this feature enhances user engagement and productivity, ensuring that users can efficiently access critical data and functionalities.

### Acceptance Criteria

- Users must be able to access the customization mode via a clearly visible 'Edit Dashboard' button or similar option.
- The interface must support drag-and-drop functionality for rearranging widgets, ensuring smooth and intuitive interactions.
- Users must be able to add new widgets from a predefined library, including widgets for market data, trading performance, risk metrics, and strategy insights.
- Users must be able to remove unwanted widgets from their dashboard with a simple action, such as dragging them to a 'Remove' area or clicking a delete icon.
- The app must save the customized layout automatically or provide a 'Save Changes' button to ensure user preferences are retained across sessions.
- The customization feature must support role-specific widgets tailored to Active Traders, Passive Investors, and Fund Managers, ensuring relevance to each user type.
- The interface must provide a preview mode for users to review their changes before finalizing the layout.
- The drag-and-drop functionality must be responsive and work seamlessly across different devices, including desktop, tablet, and mobile platforms.
- The app must include error handling to prevent issues such as widget overlap or loss of customization data during layout changes.
- The customization feature must adhere to accessibility standards, ensuring usability for users with disabilities, including keyboard navigation and screen reader compatibility.
- The app must provide a default dashboard layout option for users who prefer a pre-configured interface or wish to reset their customizations.
- Performance metrics must be monitored to ensure the drag-and-drop functionality operates smoothly without lag or delays, even with complex layouts.

- The app must include analytics to track user preferences and widget usage, enabling future improvements to the widget library and customization experience.

## Dark mode for reduced eye strain

### Description

The 'Dark mode for reduced eye strain' feature in the 'my-test-juan-0304' app provides users with an alternative visual theme designed to reduce eye strain during prolonged usage. By switching to a darker color palette, this feature enhances usability in low-light environments and improves focus by minimizing glare. It is particularly beneficial for users who spend extended periods analyzing trading data, monitoring market trends, or managing portfolios. The feature is seamlessly integrated into the app's settings, allowing users to toggle between light and dark modes based on their preferences.

### Acceptance Criteria

- The dark mode option must be accessible via the app's settings menu, with a clear toggle or switch for enabling/disabling the feature.
- The dark mode must apply consistently across all app screens, including dashboards, analytics pages, trading interfaces, and settings menus.
- Ensure that all text, icons, and interactive elements remain legible and visually distinct in dark mode, adhering to accessibility standards (WCAG AA or higher).
- Provide a preview of the dark mode theme in the settings menu, allowing users to see how the interface will look before enabling it.
- Dark mode must automatically adjust to the user's device settings if the device is set to a system-wide dark mode preference.
- Include an option for users to schedule dark mode activation based on time of day (e.g., automatically enabling dark mode during evening hours).
- Ensure compatibility with mobile and desktop versions of the app, maintaining a consistent user experience across platforms.
- Test the dark mode feature across various devices and operating systems to ensure visual consistency and performance stability.
- Provide a fallback mechanism to revert to light mode in case of any issues or user dissatisfaction with the dark mode experience.
- Monitor user feedback and analytics to assess the adoption rate and satisfaction levels of the dark mode feature, using insights to refine the implementation.

## Voice-activated navigation and commands

### Description

The 'Voice-activated navigation and commands' feature in the 'my-test-juan-0304' app enables users to interact with the platform using voice commands, providing a hands-free and efficient way to navigate and execute actions. This feature is designed to enhance accessibility and convenience, especially for users who prefer or require voice-based interaction. It supports a wide range of commands, including navigation between sections, executing trades, accessing analytics, and customizing settings. The voice recognition system is powered by advanced AI, ensuring high accuracy and responsiveness. This feature is particularly beneficial for users who multitask or operate in environments where manual interaction is limited.

### Acceptance Criteria

- The voice-activated navigation and commands feature must be accessible from the main dashboard and settings menu.
- Users must be able to activate the voice command functionality using a dedicated button or a wake word (e.g., 'Hey MyTestJuan').
- The system must support a predefined set of voice commands for core functionalities, including navigation, trade execution, analytics access, and settings customization.
- Voice recognition must demonstrate high accuracy, with a minimum success rate of 95% for supported commands in standard environments.
- The feature must support multiple languages, allowing users to interact in their preferred language.
- Provide visual feedback (e.g., confirmation messages or highlights) for recognized commands to ensure clarity and user confidence.
- Include error handling for unrecognized commands, offering suggestions or prompts to guide users toward valid commands.
- Ensure compatibility with various devices and operating systems, including mobile and desktop platforms.
- The voice command system must adhere to privacy standards, ensuring that voice data is securely processed and not stored without user consent.
- Integrate accessibility features, such as adjustable voice sensitivity and support for users with speech impairments.
- Allow users to customize voice command settings, including enabling/disabling specific commands and adjusting wake word sensitivity.
- Include a tutorial or onboarding guide to familiarize users with the voice command functionality and supported commands.
- The feature must be tested across diverse environments (e.g., noisy and quiet settings) to ensure consistent performance.
- Provide analytics to track user engagement with voice commands, enabling future improvements based on usage patterns.

## Mobile app synchronization for seamless access

### Description

The 'Mobile app synchronization for seamless access' feature in the 'my-test-juan-0304' app ensures that users can effortlessly access their trading data, strategies, and account settings across multiple devices. This feature provides real-time synchronization between the mobile app and the web platform, allowing users to switch between devices without losing progress or data. It supports automatic updates for trading activity, notifications, and portfolio changes, ensuring a consistent and uninterrupted user experience. This feature is designed to enhance accessibility and convenience for all user roles, enabling them to stay connected to their trading activities anytime, anywhere.

### Acceptance Criteria

- The mobile app must synchronize all user data, including trading strategies, portfolio details, notifications, and account settings, with the web platform in real-time.
- Synchronization must occur automatically when the user logs into the mobile app, ensuring the latest data is available without manual intervention.
- Users must be able to view and edit their trading strategies, portfolio settings, and account preferences on the mobile app, with changes reflected instantly on the web platform.
- The synchronization process must be secure, utilizing encryption protocols to protect sensitive user data during transmission.
- The feature must support offline access, allowing users to view previously synchronized data even without an active internet connection. Changes made offline should be queued and synchronized once connectivity is restored.
- Synchronization must be optimized for performance, ensuring minimal delays and efficient handling of large datasets, such as trading history and market analytics.
- The mobile app must provide visual indicators (e.g., loading icons or status messages) to inform users of synchronization progress and completion.
- Error handling mechanisms must be implemented to manage potential synchronization failures, with clear feedback provided to users and options to retry synchronization.
- The feature must be compatible with both iOS and Android devices, ensuring consistent functionality across different operating systems.
- The synchronization process must adhere to data privacy regulations, ensuring compliance with relevant laws and standards for user data protection.
- The feature must be tested across various device types and screen sizes to ensure consistent performance and usability.
- Include analytics to monitor synchronization performance and user behavior, enabling continuous improvement of the feature.

## Push notifications for trade updates and alerts

### Description

The 'Push notifications for trade updates and alerts' feature in the 'my-test-juan-0304' app ensures users stay informed about critical trading activities and market events in real-time. This feature delivers timely notifications for trade executions, price movements, risk alerts, and other relevant updates directly to users' devices. It enhances user engagement and decision-making by providing actionable insights and alerts, ensuring users can respond promptly to market changes. Notifications are customizable, allowing users to tailor the type and frequency of alerts based on their trading preferences and strategies.

### Acceptance Criteria

- Push notifications must be delivered in real-time for critical events, including trade executions, price movements, risk alerts, and market updates.
- Users must be able to customize notification preferences, including the type of alerts (e.g., trade updates, risk warnings) and frequency of notifications.
- Notifications should include relevant details, such as trade execution status, price thresholds, and risk exposure metrics, ensuring users have actionable information.
- The app must support push notifications across all platforms, including iOS, Android, and web browsers, ensuring consistent functionality for all users.
- Notifications must be visually distinct and easy to understand, with clear titles, concise messages, and optional links to detailed views within the app.
- Users must have the ability to enable or disable notifications for specific trading pairs, strategies, or risk thresholds, providing granular control over alerts.
- The app should implement a 'Do Not Disturb' mode, allowing users to temporarily pause notifications during specified time periods.
- Push notifications must be secure, ensuring sensitive trading information is encrypted and protected from unauthorized access.
- The notification system must demonstrate high reliability and performance, with minimal delays and no missed alerts, even during high market activity.
- Include error handling to manage potential failures in notification delivery, providing fallback mechanisms such as email or SMS alerts when push notifications fail.
- Notifications must adhere to accessibility standards, including support for screen readers and adjustable text sizes, ensuring usability for all users.
- Integrate analytics to track user engagement with notifications, providing insights into the effectiveness of alerts and opportunities for improvement.
- Ensure compatibility with the latest versions of mobile operating systems and devices, testing across a wide range of platforms for consistency.

## Interactive tutorials for platform features

## Description

The 'Interactive tutorials for platform features' feature in the 'my-test-juan-0304' app provides users with step-by-step guidance on how to utilize various functionalities of the platform effectively. These tutorials are designed to cater to different user roles, ensuring that Active Traders, Passive Investors, and Fund Managers can quickly learn and adapt to the tools and features relevant to their needs. The tutorials are interactive, allowing users to engage with simulated scenarios, practice workflows, and receive real-time feedback. This feature aims to enhance user onboarding, reduce learning curves, and improve overall platform engagement.

## Acceptance Criteria

- Interactive tutorials must be accessible from the main dashboard and relevant feature pages, ensuring ease of access for users.
- Tutorials should be tailored to each user role (Active Trader, Passive Investor, Fund Manager), focusing on features and workflows specific to their needs.
- The tutorials must include step-by-step instructions, visual aids (e.g., animations, tooltips), and interactive elements (e.g., clickable buttons, simulated workflows).
- Users should be able to practice key actions within the tutorials, such as placing trades, configuring strategies, or managing portfolios, in a simulated environment without affecting live data.
- Real-time feedback must be provided during tutorials, highlighting errors and offering corrective guidance to ensure users understand the processes fully.
- Tutorials should be modular, allowing users to complete them at their own pace and revisit specific sections as needed.
- Include a progress tracker within the tutorials to help users monitor their completion status and encourage engagement.
- Ensure tutorials are compatible with both desktop and mobile versions of the app, providing a consistent experience across devices.
- Tutorials must adhere to accessibility standards (e.g., WCAG) to ensure usability for users with disabilities, including support for screen readers and keyboard navigation.
- Provide multilingual support for tutorials to cater to a diverse user base, ensuring inclusivity and ease of understanding.
- Integrate analytics to track user engagement with tutorials, identifying areas for improvement and optimizing content based on user behavior.
- Include an option for users to provide feedback on tutorials, enabling continuous refinement and enhancement of the feature.
- Ensure tutorials are updated regularly to reflect changes in platform features and workflows, maintaining relevance and accuracy.

## Accessibility features for users with disabilities

### Description

The 'Accessibility features for users with disabilities' in the 'my-test-juan-0304' app ensures that the platform is inclusive and usable for all individuals, including those with disabilities. This feature incorporates accessibility standards and tools to provide an equitable experience for users with visual, auditory, motor, or cognitive impairments. It includes options such as screen reader compatibility, keyboard navigation, adjustable text sizes, high-contrast themes, and voice commands. By adhering to WCAG (Web Content Accessibility Guidelines) standards, the app aims to remove barriers and enable seamless interaction for all users, fostering inclusivity and compliance with accessibility regulations.

### Acceptance Criteria

- The app must support screen readers, ensuring that all text, buttons, and interactive elements are properly labeled and accessible.
- Keyboard navigation must be fully functional, allowing users to navigate the app without relying on a mouse or touch input.
- Users must be able to adjust text sizes and font styles to accommodate visual impairments, with options for large text and dyslexia-friendly fonts.
- High-contrast themes must be available to improve readability for users with low vision or color blindness.
- Voice commands must be implemented to allow hands-free navigation and interaction with the app, catering to users with motor impairments.
- Interactive elements, such as buttons and links, must have sufficient size and spacing to ensure ease of use for users with motor challenges.
- The app must provide captions or transcripts for any audio or video content, ensuring accessibility for users with hearing impairments.
- Accessibility settings must be easily discoverable and customizable within the app, allowing users to tailor their experience to their specific needs.
- The app must adhere to WCAG 2.1 AA standards, ensuring compliance with global accessibility guidelines.
- Regular accessibility audits must be conducted to identify and address any barriers or issues, ensuring continuous improvement of the feature.
- The app must provide feedback mechanisms for users to report accessibility challenges or suggest improvements.
- Accessibility features must be compatible across all supported devices and operating systems, ensuring consistency in user experience.

# Security & Compliance

## Secure storage and handling of exchange API keys (AES-256 encryption)

### Description

The 'Secure storage and handling of exchange API keys (AES-256 encryption)' feature in the 'my-test-juan-0304' app ensures the safe management of sensitive API keys used for integrating with cryptocurrency exchanges. This feature leverages AES-256 encryption, a robust industry-standard encryption method, to securely store API keys and prevent unauthorized access. It includes mechanisms for secure key retrieval and usage during trading operations, ensuring that keys are never exposed in plaintext. This feature is critical for maintaining user trust, safeguarding sensitive data, and complying with security best practices in the cryptocurrency trading domain.

### Acceptance Criteria

- API keys must be encrypted using AES-256 encryption before being stored in the app's database or any other storage medium.
- The app must implement secure key retrieval mechanisms, ensuring that API keys are decrypted only when required for authorized operations, such as placing trades or retrieving account balances.
- API keys must never be exposed in plaintext during any operation, including logging, debugging, or error messages.
- Access to API keys must be restricted based on user roles, ensuring that only authorized users (Active Trader, Passive Investor, Fund Manager) can manage or use their keys.
- The app must include a secure interface for users to input and update their API keys, with validation to ensure the keys are correctly formatted and functional.
- Implement measures to prevent brute force or unauthorized access attempts to the encrypted API keys, including rate limiting and anomaly detection systems.
- The app must provide users with clear instructions and warnings about the importance of safeguarding their API keys and avoiding sharing them with third parties.
- Include error handling to notify users of issues such as invalid API keys, failed encryption, or decryption errors, without exposing sensitive information.
- Regular security audits must be conducted to ensure the encryption and storage mechanisms remain robust and compliant with industry standards.
- The app must comply with relevant data protection regulations, such as GDPR or CCPA, ensuring that API keys are handled in accordance with legal requirements.
- Provide users with the ability to revoke and regenerate API keys securely within the app, ensuring that compromised keys can be replaced without disrupting trading operations.
- Ensure compatibility of the encryption and storage mechanisms across all supported devices and platforms, including mobile and desktop versions of the app.
- Integrate monitoring and logging systems to track access and usage of API keys, ensuring transparency and accountability for security-related operations.

## Multi-factor authentication system (app-based 2FA, email verification, biometric authentication)

### Description

The 'Multi-factor authentication system' feature in the 'my-test-juan-0304' app enhances account security by requiring users to verify their identity through multiple authentication methods. This system supports app-based two-factor authentication (2FA), email verification, and biometric authentication (e.g., fingerprint or facial recognition). By implementing layered security measures, the feature protects user accounts from unauthorized access and ensures compliance with industry standards for secure account management. It is designed to provide a seamless and user-friendly experience while maintaining robust security protocols.

### Acceptance Criteria

- Users must be able to enable and configure multi-factor authentication (MFA) from their account settings.
- The system must support app-based 2FA using popular authenticator apps (e.g., Google Authenticator, Authy) and generate time-based one-time passwords (TOTP).
- Email verification must be available as an alternative authentication method, requiring users to confirm their identity via a secure link sent to their registered email address.
- Biometric authentication must be supported for devices equipped with fingerprint scanners or facial recognition technology, ensuring compatibility across iOS and Android platforms.
- Users must be prompted to set up at least one MFA method during the onboarding process to enhance account security.
- The system must allow users to configure backup authentication methods (e.g., email verification) in case their primary method (e.g., app-based 2FA) becomes inaccessible.
- In cases of failed authentication attempts, the app must provide clear feedback to users and guide them through recovery options, such as resetting their MFA settings or contacting support.
- The MFA system must demonstrate high performance and reliability, ensuring quick response times for authentication requests without significant delays.



- Error handling must be implemented to manage potential issues, such as network connectivity problems or device compatibility errors, with appropriate user feedback.
- The app must log all authentication attempts (successful and failed) for compliance and monitoring purposes, ensuring transparency and accountability.
- The MFA system must adhere to industry standards for data encryption and secure storage, ensuring that sensitive information (e.g., biometric data, TOTP secrets) is protected.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The MFA interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Quarterly third-party security audits with published results

### Description

The 'Quarterly third-party security audits with published results' feature in the 'my-test-juan-0304' app ensures the platform's security and compliance by conducting regular, independent security audits. These audits are performed by reputable third-party organizations to identify vulnerabilities, assess risk, and validate the app's adherence to industry standards. The results of these audits are transparently published, providing users with confidence in the platform's security measures and commitment to safeguarding their data and assets. This feature is critical for maintaining trust, meeting regulatory requirements, and proactively addressing potential security threats.

### Acceptance Criteria

- The app must schedule and conduct security audits every quarter, performed by certified third-party security firms with expertise in cryptocurrency and trading platforms.
- Audit reports must include a detailed assessment of vulnerabilities, risk levels, and recommendations for remediation, ensuring comprehensive coverage of security aspects.
- The results of the audits must be published in a transparent manner, accessible to all users via a dedicated section in the app or website, while ensuring sensitive information is redacted to prevent exploitation.
- Users must be notified of the completion of each audit and the availability of the published results through in-app notifications, email, or other communication channels.
- The app must implement remediation plans for any identified vulnerabilities within a reasonable timeframe, with progress updates provided to users as part of the published results.
- Audit reports must comply with relevant regulatory and industry standards, such as GDPR, ISO 27001, or other applicable frameworks, ensuring legal and ethical compliance.
- The feature must include a mechanism for users to provide feedback or ask questions about the audit results, fostering transparency and engagement.
- The app must maintain a historical archive of published audit results, allowing users to review past reports and track improvements over time.
- Ensure the audit process and publication mechanism are compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The feature must demonstrate high performance and reliability, ensuring audit reports are published without delays or errors, even during high user activity periods.
- Include error handling to manage potential issues with third-party audit firms or publication processes, ensuring users are informed of any delays or disruptions.
- Integrate analytics to track user engagement with published audit results, providing insights into user concerns and areas for improvement in security communication.

## Data encryption at rest and in transit

### Description

The 'Data encryption at rest and in transit' feature in the 'my-test-juan-0304' app ensures the security and privacy of sensitive user data by implementing robust encryption mechanisms. Data stored within the app (at rest) and data transmitted between the app and external systems (in transit) are encrypted using industry-standard protocols such as AES-256 and TLS. This feature is critical for safeguarding user information, preventing unauthorized access, and maintaining compliance with cryptocurrency regulations and data protection laws.

### Acceptance Criteria

- All sensitive user data, including personal information, trading activity, and API keys, must be encrypted at rest using AES-256 encryption or an equivalent standard.
- Data transmitted between the app and external systems, such as exchanges or third-party services, must be encrypted using TLS (Transport Layer Security) 1.2 or higher to ensure secure communication.
- Encryption keys must be securely stored and managed using a key management system (KMS) to prevent unauthorized access or compromise.
- The app must implement mechanisms to detect and prevent unauthorized attempts to access encrypted data, including logging and alerting for suspicious activity.
- Encryption processes must not significantly impact the app's performance, ensuring seamless user experience during data storage and transmission.

- Regular security audits must be conducted to verify the effectiveness of encryption mechanisms and identify potential vulnerabilities.
- The app must comply with relevant data protection regulations, such as GDPR, CCPA, or other applicable laws, ensuring legal adherence to encryption standards.
- In the event of a data breach, encrypted data must remain inaccessible to unauthorized parties, demonstrating the effectiveness of encryption at rest and in transit.
- The encryption feature must be compatible with all supported devices and operating systems, ensuring consistent security across platforms.
- Provide clear documentation and user education on how encryption protects their data, enhancing trust and transparency.

## Role-based access control for sensitive features

### Description

The 'Role-based access control for sensitive features' feature in the 'my-test-juan-0304' app ensures that access to critical and sensitive functionalities is restricted based on user roles. This feature enhances security and compliance by allowing only authorized users to interact with specific features, such as advanced trading tools, risk management settings, and financial reporting. It provides a structured framework for defining permissions and access levels, ensuring that users can only access features relevant to their role. This feature is essential for maintaining data integrity, preventing unauthorized actions, and adhering to regulatory requirements.

### Acceptance Criteria

- The app must implement a role-based access control system that restricts access to sensitive features based on user roles (Active Trader, Passive Investor, Fund Manager).
- Each user role must have predefined permissions that align with their responsibilities and access needs within the app.
- Sensitive features, such as advanced trading tools, risk management settings, and financial reporting, must be accessible only to users with the appropriate role.
- The app must provide an intuitive interface for administrators to manage and customize role-based permissions, including adding, modifying, or revoking access for specific roles.
- Users attempting to access features outside their role permissions must receive a clear and friendly message explaining the restriction.
- The role-based access control system must be integrated with the app's authentication mechanisms, ensuring seamless enforcement of permissions upon login.
- Audit logs must be maintained to track access attempts and actions performed on sensitive features, providing transparency and compliance monitoring.
- The system must support dynamic updates to permissions, allowing administrators to adjust access levels as needed without requiring downtime.
- Role-based access control must be tested across all user roles to ensure consistent functionality and security.
- The feature must comply with relevant security standards and regulations, such as GDPR, ensuring user data privacy and protection.
- The app must include error handling to manage potential issues with role-based access control, such as incorrect permissions or system failures, providing feedback to users and administrators.
- The role-based access control system must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.

## Blockchain-based data integrity and audit trails

### Description

The 'Blockchain-based data integrity and audit trails' feature in the 'my-test-juan-0304' app leverages blockchain technology to ensure the immutability and transparency of critical trading and user activity data. By recording transactions, trade executions, and system events on a secure blockchain ledger, this feature provides users with a tamper-proof audit trail for compliance and accountability. It enhances trust by enabling users to verify the authenticity of data and ensures regulatory adherence by maintaining a transparent record of all activities. This feature is particularly valuable for fund managers and traders who require robust mechanisms for tracking and validating their actions within the platform.

### Acceptance Criteria

- All critical trading activities, including trade executions, strategy changes, and user actions, must be recorded on a secure blockchain ledger.
- The blockchain ledger must ensure immutability, preventing unauthorized modifications or deletions of recorded data.
- Users must be able to access and view their activity logs and audit trails through a dedicated interface within the app.
- The audit trail must include detailed information such as timestamps, user IDs, transaction IDs, and relevant metadata for each recorded event.
- The blockchain implementation must comply with relevant regulations and standards for data security and integrity in the cryptocurrency and trading industry.
- Fund managers must have access to aggregated audit trails for accounts they manage, enabling them to monitor and validate activities across multiple users.
- The feature must support real-time recording of events to ensure up-to-date audit trails without delays.

- The blockchain ledger must be scalable to handle high volumes of data without compromising performance or reliability.
- The app must provide mechanisms for users to verify the authenticity of data recorded on the blockchain, such as cryptographic proofs or hash comparisons.
- In case of disputes or compliance audits, the blockchain-based audit trail must serve as a reliable source of truth for resolving issues.
- The feature must include error handling and fallback mechanisms to ensure data integrity in cases of network disruptions or blockchain node failures.
- The blockchain ledger must be encrypted and secured against unauthorized access, adhering to industry best practices for blockchain security.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The audit trail interface must be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards to ensure usability for all users.

## Real-time threat detection and response systems

### Description

The 'Real-time threat detection and response systems' feature in the 'my-test-juan-0304' app is designed to proactively identify and mitigate security threats as they occur. Leveraging advanced algorithms and machine learning, this feature continuously monitors user activity, system logs, and external data sources to detect anomalies, unauthorized access attempts, and potential vulnerabilities. Upon identifying a threat, the system initiates immediate countermeasures, such as isolating affected accounts, notifying users, and providing actionable recommendations to secure their assets. This feature is critical for ensuring the safety of user data, trading activities, and overall platform integrity.

### Acceptance Criteria

- The system must continuously monitor user activity, API interactions, and system logs for potential security threats in real-time.
- Threat detection algorithms must utilize machine learning and pattern recognition to identify anomalies, unauthorized access attempts, and suspicious activities.
- Upon detecting a threat, the system must automatically initiate countermeasures, such as isolating affected accounts, blocking suspicious IP addresses, or temporarily pausing trading activities.
- Users must receive immediate notifications via in-app alerts, email, and SMS when a threat is detected, including details of the issue and recommended actions.
- The system must provide a detailed threat report accessible via the user dashboard, outlining the nature of the threat, actions taken, and preventive measures for future incidents.
- The feature must integrate seamlessly with other security measures, such as multi-factor authentication and encryption protocols, to enhance overall platform security.
- The system must demonstrate high performance and reliability, ensuring minimal false positives and quick response times to genuine threats.
- Threat detection capabilities must be regularly updated to adapt to emerging security risks and vulnerabilities in the cryptocurrency and trading ecosystem.
- The feature must comply with relevant security and data protection regulations, ensuring user privacy and legal adherence.
- The system must include a feedback mechanism for users to report false positives or additional security concerns, enabling continuous improvement of threat detection algorithms.
- The feature must be compatible across all devices and operating systems supported by the app, ensuring consistent functionality for all users.
- The system must provide role-specific threat mitigation options, such as tailored recommendations for Active Traders, Passive Investors, and Fund Managers based on their trading activities and account configurations.

## Compliance with relevant cryptocurrency regulations

### Description

The 'Compliance with relevant cryptocurrency regulations' feature in the 'my-test-juan-0304' app ensures adherence to applicable legal and regulatory requirements in the cryptocurrency trading space. This feature is designed to provide users with a secure and compliant trading environment by implementing measures that align with global and regional cryptocurrency regulations. It includes automated checks for regulatory compliance, reporting tools for audits, and mechanisms to ensure transparency in trading activities. By integrating compliance protocols, the app minimizes legal risks for users and enhances trustworthiness, making it suitable for both individual traders and institutional fund managers.

### Acceptance Criteria

- The app must implement automated checks to ensure compliance with major cryptocurrency regulations, including AML (Anti-Money Laundering) and KYC (Know Your Customer) requirements.
- Users must be able to submit necessary identification documents for KYC verification directly through the app, with secure storage and encryption of sensitive data.
- The app should provide real-time monitoring and alerts for suspicious activities, such as unusual trading patterns or large transactions, to ensure adherence to AML protocols.

- Compliance reports must be generated automatically and made accessible to users, including audit trails for trading activities and fund movements.
- The app must integrate with regulatory reporting systems to facilitate seamless submission of required data to authorities, where applicable.
- Ensure compatibility with region-specific cryptocurrency regulations, including GDPR (General Data Protection Regulation) for EU users and other relevant laws for different jurisdictions.
- Provide users with clear and transparent information about the app's compliance measures, including terms of service and privacy policies, to build trust and confidence.
- Include a mechanism for users to report potential compliance issues or concerns directly through the app, with a dedicated support team to address these reports promptly.
- The app must undergo regular third-party audits to verify compliance with cryptocurrency regulations, with published results available to users for transparency.
- Ensure that compliance measures do not compromise user experience or trading performance, maintaining a balance between security and usability.
- The app must provide educational resources and updates about changes in cryptocurrency regulations, helping users stay informed and compliant.
- Implement role-based compliance features, such as enhanced reporting tools for Fund Managers and simplified compliance workflows for Passive Investors and Active Traders.

## Anomaly detection systems for proactive security measures

### Description

The 'Anomaly detection systems for proactive security measures' feature in the 'my-test-juan-0304' app is designed to enhance the security of user accounts and trading activities by identifying and addressing unusual patterns or behaviors in real-time. Leveraging advanced machine learning algorithms, this feature continuously monitors user activity, trading patterns, and system interactions to detect anomalies such as unauthorized access attempts, irregular trading volumes, or unusual API usage. Upon detecting an anomaly, the system triggers alerts and initiates predefined security protocols to mitigate potential risks. This feature is essential for safeguarding user assets, ensuring compliance, and maintaining trust in the platform.

### Acceptance Criteria

- The anomaly detection system must continuously monitor user activity, trading patterns, and API interactions in real-time.
- The system should utilize machine learning algorithms to identify unusual behaviors, such as unauthorized access attempts, irregular trading volumes, or unexpected API usage.
- Upon detecting an anomaly, the system must trigger immediate alerts via in-app notifications, email, and SMS to inform the affected user and relevant administrators.
- The system should automatically initiate predefined security protocols, such as temporarily locking the affected account, pausing trading activities, or requiring additional authentication steps.
- Users must have access to a detailed report of detected anomalies, including timestamps, descriptions, and recommended actions, through the app's security dashboard.
- The anomaly detection system must integrate seamlessly with other security features, such as multi-factor authentication and role-based access control, to provide a comprehensive security framework.
- The system should support customization options for users and administrators to define thresholds and parameters for anomaly detection based on their specific needs and risk tolerance.
- The feature must demonstrate high accuracy in distinguishing between legitimate activities and potential threats, minimizing false positives and negatives.
- The system should comply with relevant cryptocurrency regulations and data privacy standards, ensuring secure handling of user data and activity logs.
- Performance metrics, such as detection speed and accuracy, must be regularly monitored and optimized to ensure the system remains effective as the platform scales.
- The anomaly detection system must be compatible with all user roles (Active Trader, Passive Investor, Fund Manager) and adapt to their unique activity patterns and security requirements.
- The feature should be tested across different devices and operating systems to ensure consistent functionality and reliability.

## Secure payment processing for subscription and performance fees

### Description

The 'Secure payment processing for subscription and performance fees' feature in the 'my-test-juan-0304' app ensures a seamless and secure mechanism for handling user payments. This feature supports multiple payment methods, including credit/debit cards, bank transfers, and cryptocurrency, providing flexibility to users. It incorporates advanced encryption protocols (AES-256) to safeguard sensitive payment information and complies with industry standards for secure transactions. The feature also includes automated invoicing, payment reminders, and transparent fee breakdowns, enhancing user trust and simplifying financial management. By ensuring secure and efficient payment processing, this feature is critical for maintaining user confidence and supporting the app's subscription-based and performance fee models.

## Acceptance Criteria

- The payment processing system must support multiple payment methods, including credit/debit cards, bank transfers, and cryptocurrency, ensuring flexibility for users.
- All payment transactions must be encrypted using AES-256 encryption to protect sensitive user data and comply with industry security standards.
- The system must provide automated invoicing and payment reminders to users, ensuring timely payments and reducing manual effort.
- A transparent fee breakdown must be displayed to users before completing a transaction, detailing subscription fees, performance fees, and any applicable taxes or charges.
- The payment processing system must comply with relevant financial regulations and standards, including PCI DSS (Payment Card Industry Data Security Standard) for card payments.
- Users must be able to view their payment history and download invoices directly from the app for record-keeping and compliance purposes.
- The system must include error handling mechanisms to manage failed transactions, providing clear feedback to users and allowing them to retry or choose alternative payment methods.
- Performance fees must be calculated accurately based on the high-water mark tracking model, ensuring fairness and transparency for users.
- The payment processing system must integrate seamlessly with the app's subscription management module, allowing users to upgrade, downgrade, or cancel their plans without issues.
- Real-time notifications must be sent to users upon successful payment processing, ensuring they are informed of transaction status.
- The system must be tested across different devices and operating systems to ensure consistent functionality and user experience.
- The payment interface must adhere to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.
- The system must include fraud detection mechanisms to identify and prevent unauthorized transactions, enhancing security for users.

# Mobile App

## Cross-platform mobile app for iOS and Android

### Description

The 'Cross-platform mobile app for iOS and Android' feature in the 'my-test-juan-0304' app ensures seamless access to trading functionalities and insights across both major mobile operating systems. This feature provides users with a consistent and optimized experience, enabling them to monitor market activity, execute trades, and manage portfolios on-the-go. The app is designed to cater to the diverse needs of Active Traders, Passive Investors, and Fund Managers, offering intuitive navigation, real-time updates, and robust security measures. With offline access and mobile-specific enhancements, this feature empowers users to stay connected to their trading activities anytime, anywhere.

### Acceptance Criteria

- The mobile app must be fully functional and compatible with both iOS and Android platforms, supporting the latest versions of each operating system.
- Users must be able to log in securely using multi-factor authentication, including biometric options such as fingerprint or facial recognition.
- The app should provide real-time updates on market activity, portfolio performance, and trade execution, ensuring users have access to the latest information.
- The user interface must be optimized for mobile devices, offering intuitive navigation and responsive design for various screen sizes.
- Offline access must be available for viewing trading data and strategies, with synchronization occurring automatically when the device reconnects to the internet.
- Push notifications must be implemented to alert users about trade updates, risk alerts, and other critical events, with customizable notification preferences.
- The app should include augmented reality features for market visualization, providing innovative ways to analyze data and trends.
- Voice commands must be supported for hands-free operation, allowing users to perform key actions such as searching for market data or executing trades using voice input.
- Integration with smartwatches and wearable devices must be available, enabling users to receive notifications and monitor key metrics directly from their wearable devices.
- Interactive tutorials must be included to guide users through mobile app navigation and features, ensuring a smooth onboarding experience.
- The app must demonstrate high performance and reliability, with quick response times and minimal crashes or bugs during usage.
- Robust security measures, including data encryption and secure storage of sensitive information, must be implemented to protect user data and trading activities.
- The app must adhere to accessibility standards, ensuring usability for users with disabilities, including support for screen readers and adjustable text sizes.

## Offline access to trading data and strategies

### Description

The 'Offline access to trading data and strategies' feature in the 'my-test-juan-0304' app allows users to access critical trading data and strategies even when they are not connected to the internet. This ensures uninterrupted access to essential information, enabling users to review their trading plans, analyze historical data, and prepare for market activity without relying on real-time connectivity. The feature is designed to enhance user convenience and reliability, especially in scenarios where internet access is limited or unavailable.

### Acceptance Criteria

- Users must be able to access previously downloaded trading data and strategies without an active internet connection.
- The app should provide an option for users to manually download specific datasets, strategies, and reports for offline use.
- Offline data must include historical trade performance metrics, strategy templates, and market analytics that were previously synced while online.
- The app should display a clear indicator for offline mode, ensuring users are aware they are viewing cached data rather than real-time updates.
- Data synchronization must occur automatically when the user regains internet connectivity, ensuring offline data is updated with the latest information.
- The offline feature must support all major functionalities related to data review and strategy analysis, excluding real-time market updates and trade execution.
- Ensure data security by encrypting offline data storage using AES-256 encryption to protect sensitive information from unauthorized access.
- The app must provide a notification or alert if the user attempts to perform actions requiring internet connectivity while in offline mode.
- Offline access must be optimized for performance, ensuring quick loading times and smooth navigation of cached data and strategies.
- The feature must be compatible with both iOS and Android devices, ensuring consistent functionality across platforms.
- Include error handling to manage scenarios where offline data is corrupted or unavailable, providing users with clear feedback and recovery options.

- The offline access interface must adhere to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Push notifications for trade updates and risk alerts

### Description

The 'Push notifications for trade updates and risk alerts' feature in the 'my-test-juan-0304' app ensures users stay informed about critical trading activities and risk-related events in real-time. This feature delivers timely notifications for trade executions, market movements, and risk thresholds, enabling users to make informed decisions and take immediate action when necessary. Notifications are customizable, allowing users to tailor alerts based on their trading preferences and risk tolerance. The feature is designed to enhance user engagement and responsiveness, ensuring that users never miss important updates or opportunities.

### Acceptance Criteria

- Push notifications must be delivered in real-time for trade updates, including trade executions, order placements, and cancellations.
- Risk alerts must notify users when predefined thresholds are breached, such as maximum drawdown, stop-loss triggers, or circuit breaker activations.
- Users must be able to customize notification preferences, including the type of alerts they wish to receive (e.g., trade updates, risk alerts) and the frequency of notifications.
- Notifications must be accessible across all supported devices, including mobile apps, smartwatches, and wearable devices, ensuring seamless user experience.
- The app must provide a clear and intuitive interface for managing notification settings, allowing users to enable, disable, or modify alerts easily.
- Push notifications must include relevant details, such as trade symbols, execution prices, risk metrics, and timestamps, ensuring users have actionable information at a glance.
- Notifications must be delivered securely, adhering to data privacy standards and ensuring sensitive trading information is not exposed.
- The app must implement a fallback mechanism to ensure notifications are delivered via email or SMS in case of app connectivity issues.
- Users must be able to view a history of past notifications within the app for reference and tracking purposes.
- The notification system must demonstrate high reliability and performance, with minimal delays and no missed alerts during high market activity periods.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Push notifications must adhere to accessibility standards, ensuring they are usable by all users, including those with disabilities (e.g., text-to-speech support for visually impaired users).
- The app must provide analytics for notification engagement, allowing administrators to track user interaction with alerts and optimize the notification system.

## Mobile-optimized user interface for all features

### Description

The 'Mobile-optimized user interface for all features' in the 'my-test-juan-0304' app ensures seamless access to all functionalities on mobile devices. This feature is designed to provide an intuitive and responsive interface that adapts to various screen sizes and resolutions, enabling users to efficiently navigate and utilize the app's trading, analytics, and management tools on the go. It prioritizes usability and performance, ensuring that users can perform critical tasks such as monitoring trades, managing portfolios, and accessing reports without compromising functionality or user experience.

### Acceptance Criteria

- The mobile interface must be fully responsive, adapting to different screen sizes and resolutions, including smartphones and tablets.
- All core features of the app, including trading, analytics, reporting, and portfolio management, must be accessible and functional on the mobile interface.
- The design must prioritize ease of navigation, with intuitive menus, buttons, and gestures optimized for touchscreens.
- Performance on mobile devices must be comparable to the desktop version, with minimal loading times and smooth transitions between screens.
- The mobile interface must support both portrait and landscape orientations, allowing users to choose their preferred viewing mode.
- Interactive elements, such as charts and dashboards, must be optimized for touch input, enabling users to zoom, scroll, and interact seamlessly.
- Accessibility features, such as larger text options and voice commands, must be integrated into the mobile interface to accommodate users with disabilities.
- The mobile interface must be compatible with the latest versions of iOS and Android operating systems, ensuring broad device support.
- Error handling mechanisms must be implemented to provide clear feedback to users in case of connectivity issues or unexpected errors.
- The mobile interface must include a dark mode option to reduce eye strain during extended use.

- Security features, such as biometric authentication and encrypted data transmission, must be seamlessly integrated into the mobile interface.
- The mobile interface must undergo rigorous testing across various devices and operating systems to ensure consistency and reliability.
- User feedback mechanisms must be included to allow users to report issues or suggest improvements for the mobile interface.

## Biometric authentication for enhanced security

### Description

The 'Biometric authentication for enhanced security' feature in the 'my-test-juan-0304' app provides an advanced layer of security by leveraging biometric data such as fingerprints or facial recognition. This feature ensures that only authorized users can access their accounts, reducing the risk of unauthorized access and enhancing overall platform security. It integrates seamlessly with modern devices that support biometric authentication, offering a convenient and secure login experience. This feature is particularly beneficial for users managing sensitive financial data and executing trades, ensuring their accounts remain protected.

### Acceptance Criteria

- Biometric authentication must be available for devices that support fingerprint scanning, facial recognition, or other biometric methods.
- Users must be able to enable or disable biometric authentication through the app's security settings.
- Biometric authentication must work in conjunction with multi-factor authentication, providing an additional layer of security.
- The app must securely store and handle biometric data, adhering to industry standards and regulations for data privacy and security.
- Biometric authentication must demonstrate high reliability and accuracy, minimizing false positives and negatives during login attempts.
- In cases where biometric authentication fails or is unavailable (e.g., device compatibility issues), users must be able to log in using alternative methods such as passwords or PINs.
- The feature must be compatible with both iOS and Android devices, ensuring cross-platform functionality.
- Biometric authentication must integrate seamlessly with other app features, such as secure payment processing and sensitive data access.
- The app must provide clear instructions and feedback during the biometric setup process, ensuring users can easily configure the feature.
- The feature must comply with relevant security and privacy regulations, such as GDPR, CCPA, and other applicable standards.
- Biometric authentication must be tested across a variety of devices and operating systems to ensure consistent performance and compatibility.
- The app must include error handling and user notifications for scenarios where biometric authentication encounters issues, such as hardware malfunctions or software errors.
- Analytics must be integrated to monitor the usage and performance of biometric authentication, enabling continuous improvement of the feature.

## Augmented reality features for market visualization

### Description

The 'Augmented reality features for market visualization' in the 'my-test-juan-0304' app provides users with an innovative way to interact with market data using augmented reality (AR) technology. This feature enables users to visualize complex trading metrics, market trends, and portfolio performance in a 3D AR environment, offering an immersive and intuitive experience. By overlaying real-time data onto physical spaces, users can gain deeper insights into market dynamics and make informed trading decisions. This feature is designed to enhance decision-making for advanced users by presenting data in a visually engaging and interactive format.

### Acceptance Criteria

- The augmented reality feature must be accessible through the mobile app on devices that support AR technology, such as smartphones and tablets with ARKit (iOS) or ARCore (Android).
- Users must be able to activate the AR mode from the main dashboard or a dedicated menu option within the app.
- The AR visualization should include key trading metrics such as price movements, volume, market trends, and portfolio performance, displayed in an interactive 3D format.
- Users must be able to customize the AR view by selecting specific data points, timeframes, or trading pairs to focus on.
- The AR environment should support real-time updates, ensuring that displayed data is accurate and reflects current market conditions.
- Interactive elements, such as touch gestures or voice commands, must allow users to navigate, zoom, and filter data within the AR visualization.
- The feature should include a tutorial or onboarding guide to help users understand how to use AR for market visualization effectively.
- Ensure compatibility with various device screen sizes and resolutions, maintaining a consistent and high-quality user experience.
- The AR feature must demonstrate high performance, with minimal latency or lag during data rendering and interaction.
- Include error handling to manage scenarios where AR functionality is unavailable due to device limitations or permissions, providing fallback options for data visualization.



- The AR visualization must adhere to accessibility standards, ensuring usability for users with disabilities, including support for voice commands and adjustable text sizes.
- Integrate analytics to track user engagement with the AR feature, providing insights for future improvements and optimizations.
- Ensure the AR feature complies with relevant data privacy and security standards, protecting sensitive user information during data visualization.
- The feature must be thoroughly tested across supported devices and operating systems to ensure reliability and consistency.

## Voice commands for hands-free operation

### Description

The 'Voice commands for hands-free operation' feature in the 'my-test-juan-0304' app enables users to interact with the platform using voice commands, providing a convenient and efficient way to navigate and execute actions without manual input. This feature is designed to enhance accessibility and usability, especially for users who prefer or require hands-free operation due to multitasking or physical limitations. It supports a wide range of commands, including navigation, trade execution, portfolio management, and accessing analytics, ensuring seamless functionality across the app.

### Acceptance Criteria

- The voice command functionality must be accessible from the main dashboard and other key sections of the app.
- Users must be able to activate voice commands using a dedicated button or a wake word (e.g., 'Hey MyTest').
- The feature should support common actions such as navigating to specific sections (e.g., 'Go to portfolio'), executing trades (e.g., 'Buy 10 shares of BTC'), and accessing analytics (e.g., 'Show performance dashboard').
- Voice recognition must demonstrate high accuracy, with support for multiple accents and languages to accommodate diverse user demographics.
- The app should provide visual and auditory feedback to confirm recognized commands and executed actions.
- Include a help section or tutorial to guide users on available voice commands and their usage.
- Ensure compatibility with mobile devices, smartwatches, and other wearable devices that support voice input.
- The feature must adhere to privacy standards, ensuring that voice data is securely processed and not stored without user consent.
- Implement error handling for unrecognized commands, providing suggestions or prompts to guide users toward successful interactions.
- Voice commands must function reliably in real-time, with minimal latency to ensure a smooth user experience.
- The feature should be tested across different devices and operating systems to ensure consistent performance.
- Accessibility standards must be met, ensuring usability for users with disabilities, including those with visual or motor impairments.
- Integrate analytics to track voice command usage patterns, enabling continuous improvement of the feature.

## Integration with smartwatches and wearable devices

### Description

The 'Integration with smartwatches and wearable devices' feature in the 'my-test-juan-0304' app enables users to access critical trading information and alerts directly on their wearable devices, such as smartwatches. This feature is designed to provide real-time updates, notifications, and simplified interaction for users who are on the move or prefer hands-free access to trading data. It enhances convenience and responsiveness by allowing users to monitor market activity, receive risk alerts, and execute basic trading actions without needing to access their mobile phones or computers. The feature supports seamless synchronization with the app and ensures a user-friendly interface optimized for smaller screens.

### Acceptance Criteria

- The app must support integration with popular smartwatch platforms, including Apple Watch, Wear OS, and other major wearable device ecosystems.
- Users must be able to receive real-time notifications for trade updates, risk alerts, and market activity directly on their wearable devices.
- The smartwatch interface must display essential trading metrics, such as portfolio performance, current exposure, and market trends, in a visually clear and concise manner.
- Users must be able to execute basic trading actions, such as accepting or rejecting trade recommendations, directly from their wearable devices.
- The feature must include customizable notification preferences, allowing users to choose which types of alerts they want to receive on their wearable devices.
- The smartwatch interface must be optimized for smaller screens, ensuring readability and usability without overwhelming the user with excessive information.
- The integration must support biometric authentication (if available on the wearable device) to ensure secure access to trading data and actions.
- The feature must demonstrate high performance and reliability, with minimal latency in delivering notifications and updates to wearable devices.
- Error handling must be implemented to manage potential connectivity issues between the app and wearable devices, providing users with clear feedback and troubleshooting options.

- The integration must adhere to data privacy and security standards, ensuring that sensitive trading information is encrypted and protected during transmission to wearable devices.
- The feature must be compatible with the latest version of the app and tested across different wearable devices and operating systems for consistency.
- Interactive tutorials must be provided to guide users on how to set up and use the smartwatch integration effectively.
- The smartwatch interface must support accessibility features, such as voice commands and haptic feedback, to ensure usability for all users, including those with disabilities.

## Interactive tutorials for mobile app navigation

### Description

The 'Interactive tutorials for mobile app navigation' feature in 'my-test-juan-0304' app provides users with step-by-step guidance to familiarize themselves with the mobile app's functionalities. Designed to cater to all user roles, the tutorials offer an engaging and hands-on learning experience, ensuring users can efficiently navigate the app and utilize its features. The tutorials are tailored to the specific needs of Active Traders, Passive Investors, and Fund Managers, highlighting relevant tools and workflows. This feature enhances user onboarding and reduces the learning curve, enabling users to maximize the app's potential.

### Acceptance Criteria

- The tutorials must be accessible from the mobile app's main menu or onboarding screen, ensuring visibility and ease of access.
- Interactive tutorials should include visual aids, such as animations, tooltips, and step-by-step instructions, to guide users through key app functionalities.
- Tutorials must be role-specific, providing tailored content for Active Traders, Passive Investors, and Fund Managers, focusing on features relevant to their needs.
- Users should have the option to skip or revisit tutorials at any time, allowing flexibility based on their familiarity with the app.
- The tutorials must cover essential app features, including navigation, trading tools, analytics dashboards, risk management settings, and subscription management.
- Interactive elements, such as quizzes or checkpoints, should be included to ensure users understand the content before proceeding to the next step.
- The tutorials must be optimized for mobile devices, ensuring smooth performance and responsiveness across iOS and Android platforms.
- Tutorials should support multiple languages to accommodate a diverse user base, with localization for key markets.
- The app must track user progress within the tutorials, allowing users to resume from where they left off if interrupted.
- Accessibility features, such as voice narration, adjustable text sizes, and high-contrast modes, must be integrated to ensure usability for users with disabilities.
- The tutorials must be updated regularly to reflect changes in app features or workflows, ensuring content remains relevant and accurate.
- Feedback mechanisms, such as user ratings or comments, should be included to gather insights for improving the tutorials.
- Analytics should be implemented to track user engagement with the tutorials, providing data on completion rates and areas where users may struggle.

## Customizable notification preferences for users

### Description

The 'Customizable notification preferences for users' feature in the 'my-test-juan-0304' app empowers users to tailor their notification settings according to their individual needs and preferences. This feature allows users to select the types of notifications they wish to receive, such as trade updates, risk alerts, market news, or subscription reminders. Users can also customize the delivery method (e.g., in-app, email, SMS, or push notifications) and set specific time windows for receiving notifications to avoid disruptions. By providing granular control over notification settings, this feature enhances user experience, ensuring that users stay informed without being overwhelmed by irrelevant or untimely alerts.

### Acceptance Criteria

- Users must be able to access the notification preferences settings from their profile or settings menu within the app.
- The notification preferences interface should be intuitive and user-friendly, allowing users to easily enable or disable specific types of notifications (e.g., trade updates, risk alerts, market news).
- Users must be able to select their preferred delivery method for notifications, including in-app notifications, email, SMS, and push notifications.
- The app should allow users to set specific time windows during which notifications can be delivered, ensuring flexibility and minimizing disruptions.
- Users must be able to customize notification frequency (e.g., real-time, daily summary, weekly summary) for certain types of alerts, such as market news or performance updates.
- The app should provide a preview or test notification option to help users verify their settings before saving changes.
- Notification preferences must be saved and applied immediately, with changes reflected across all devices linked to the user's account.
- The app should include default notification settings for new users, which can be customized later based on individual preferences.

- Ensure that notifications related to critical events, such as risk alerts or account security issues, cannot be completely disabled but can be adjusted in terms of delivery method and frequency.
- The notification system must demonstrate high reliability, ensuring timely delivery of alerts without significant delays or failures.
- Include error handling to manage potential issues with notification delivery, such as invalid email addresses or phone numbers, and provide feedback to users about the error.
- The notification preferences feature must adhere to data privacy regulations, ensuring that user contact information is securely stored and used only for authorized purposes.
- Integrate analytics to track user engagement with notifications, providing insights for further optimization of the notification system.
- Ensure the notification preferences feature is compatible with the latest version of the app and is tested across different devices and operating systems for consistency.
- The notification preferences interface should be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

# Subscription & Pricing

## Multiple subscription tiers with differentiated features and limits

### Description

The 'Multiple subscription tiers with differentiated features and limits' feature in the 'my-test-juan-0304' app provides users with a variety of subscription plans tailored to their specific needs and trading preferences. Each tier offers a unique set of features, limits, and benefits, ensuring that users can select a plan that aligns with their trading goals and budget. This feature is designed to accommodate the diverse requirements of Active Traders, Passive Investors, and Fund Managers, offering flexibility and scalability as users grow in their trading journey. The subscription tiers are transparently presented, allowing users to make informed decisions about their plan. Additionally, the feature includes mechanisms for upgrading, downgrading, or customizing plans to suit evolving needs.

### Acceptance Criteria

- The app must provide at least three distinct subscription tiers (e.g., Basic, Pro, Enterprise), each with clearly defined features, limits, and pricing.
- Users must be able to view a detailed comparison of subscription tiers, including features such as trading limits, access to advanced analytics, AI-driven insights, and priority support.
- The subscription tiers must cater to all user roles (Active Trader, Passive Investor, Fund Manager), with specific features tailored to each role's needs.
- Users must be able to upgrade or downgrade their subscription tier seamlessly within the app, with changes taking effect immediately or at the start of the next billing cycle.
- The app must support multiple payment methods, including credit cards, cryptocurrency, and other secure payment options, for subscription purchases.
- A free trial period must be available for new users, allowing them to explore the features of a higher-tier plan before committing to a subscription.
- Discounts for long-term subscriptions (e.g., annual plans) must be clearly displayed and applied during checkout.
- Enterprise users must have the option to customize their subscription plan, including tailored features and limits, with support from the app's customer service team.
- The app must provide automated invoicing and payment reminders to ensure users are informed about upcoming charges and subscription renewals.
- Users must be able to cancel their subscription at any time, with clear instructions and no hidden fees or penalties.
- The subscription management interface must be intuitive and accessible, allowing users to easily view their current plan, billing history, and available upgrades or downgrades.
- The app must include a transparent fee breakdown, showing users exactly what they are paying for and any additional charges (e.g., performance-based fees).
- AI-driven recommendations must be integrated to suggest the optimal subscription tier based on the user's trading activity, preferences, and goals.
- The subscription feature must comply with relevant financial regulations and data privacy standards, ensuring secure handling of payment information and user data.

## Performance-based fee structure with high-water mark tracking

### Description

The 'Performance-based fee structure with high-water mark tracking' feature in the 'my-test-juan-0304' app ensures a fair and transparent pricing model for users. This feature calculates fees based on the user's trading performance, incentivizing better results while protecting users from being charged during periods of underperformance. The high-water mark mechanism ensures that fees are only applied when the user's portfolio value exceeds its previous peak, preventing double-charging for the same gains. This feature is particularly beneficial for fund managers and active traders, aligning the app's pricing structure with user success and fostering trust.

### Acceptance Criteria

- The app must calculate fees based on the user's portfolio performance, using a percentage of profits as the basis for the fee structure.
- Implement a high-water mark mechanism to ensure fees are only charged when the portfolio value exceeds its previous highest value.
- Provide a detailed breakdown of performance-based fees, including the calculation method and the high-water mark threshold, in the user's dashboard.
- Ensure the high-water mark tracking is persistent across sessions and accurately reflects the user's portfolio history.
- Include visual indicators in the dashboard to show the current portfolio value, previous high-water mark, and potential fees if gains are realized.
- Support different fee percentages for various subscription tiers, allowing customization based on user preferences and account type.
- Enable fund managers to view aggregated performance-based fees for all managed accounts, with individual account-level details available on request.

- Provide notifications to users when their portfolio value approaches or exceeds the high-water mark, along with an explanation of potential fees.
- Ensure the fee structure complies with relevant financial regulations and standards, including transparency and user consent for fee deductions.
- Include error handling to address discrepancies in portfolio value calculations or high-water mark tracking, with clear communication to users about any issues.
- Integrate analytics to track user portfolio performance and fee trends, providing insights for optimizing the fee structure and improving user satisfaction.
- Ensure compatibility with all subscription tiers and user roles, with tailored fee structures for Active Traders, Passive Investors, and Fund Managers.
- Test the feature across different devices and operating systems to ensure consistent functionality and user experience.
- Adhere to data security standards, ensuring that sensitive financial information related to portfolio performance and fees is encrypted and protected.

## Secure billing system supporting multiple payment methods (including cryptocurrency)

### Description

The 'Secure billing system supporting multiple payment methods (including cryptocurrency)' feature in the 'my-test-juan-0304' app ensures a seamless and secure payment experience for users subscribing to various tiers of the platform. This feature supports multiple payment methods, including credit/debit cards, bank transfers, and popular cryptocurrencies, catering to diverse user preferences. It incorporates robust security measures, such as encryption and fraud detection, to safeguard sensitive payment information. The system is designed to provide transparency in billing, allowing users to view detailed fee breakdowns and track payment history. Additionally, it supports automated invoicing and payment reminders, ensuring users stay informed about their subscription status.

### Acceptance Criteria

- The billing system must support multiple payment methods, including credit/debit cards, bank transfers, and at least three popular cryptocurrencies (e.g., Bitcoin, Ethereum, USDT).
- Payment processing must be secure, utilizing AES-256 encryption for data at rest and TLS encryption for data in transit.
- The system must comply with relevant financial regulations and standards, such as PCI DSS (Payment Card Industry Data Security Standard) for card payments.
- Users must be able to view a detailed breakdown of subscription fees, including base fees, performance-based fees, and any applicable discounts or taxes.
- The system must provide automated invoicing, generating clear and professional invoices for each payment transaction.
- Payment reminders must be sent to users via email and in-app notifications, with customizable frequency settings.
- The billing system must include a transaction history feature, allowing users to review past payments and invoices.
- Cryptocurrency payments must be processed securely, with real-time exchange rate calculations and confirmation of transactions on the blockchain.
- The system must include fraud detection mechanisms to identify and prevent unauthorized transactions or suspicious activity.
- Users must be able to update their payment methods and billing information easily through the app's settings.
- The billing system must integrate with financial software for enterprise users, enabling streamlined subscription management and reporting.
- The system must demonstrate high performance and reliability, ensuring payments are processed without delays or errors.
- Error handling must be implemented to provide clear feedback to users in case of payment failures, including retry options and support contact information.
- The billing interface must be user-friendly and accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards.

## Free trial period for new users

### Description

The 'Free trial period for new users' feature in the 'my-test-juan-0304' app allows new users to explore and experience the platform's functionalities without any upfront payment. This feature is designed to provide users with a risk-free opportunity to evaluate the app's capabilities, including trading strategies, analytics, AI-driven insights, and more. The free trial period is time-limited and includes access to a predefined set of features based on the user's role (Active Trader, Passive Investor, or Fund Manager). By offering a free trial, the app aims to increase user adoption and engagement while demonstrating its value proposition.

### Acceptance Criteria

- The free trial period must be clearly communicated to new users during the sign-up process, including its duration and the features available.
- Users must be able to activate the free trial without requiring payment information upfront, ensuring a seamless onboarding experience.
- The free trial should provide access to core features relevant to the user's role, such as trading strategies for Active Traders, portfolio optimization tools for Passive Investors, and bulk user management for Fund Managers.

- A countdown timer or notification system must be implemented to inform users of the remaining time in their free trial period.
- Upon expiration of the free trial, users should receive a prompt to select a subscription tier, with clear details about pricing and features.
- The app must ensure that users on the free trial cannot access premium features reserved for paid subscription tiers, maintaining differentiation between free and paid plans.
- Analytics should be integrated to track user engagement and feature usage during the free trial, providing insights for improving the onboarding experience.
- The free trial feature must comply with relevant regulations, including data privacy and subscription management laws, ensuring transparency and user trust.
- Error handling must be implemented to address potential issues during the activation or expiration of the free trial, providing clear feedback to users.
- The free trial feature must be compatible with all supported devices and platforms, including the mobile app, ensuring consistency across user experiences.

## Discounts for long-term subscriptions

### Description

The 'Discounts for long-term subscriptions' feature in the 'my-test-juan-0304' app provides users with cost-saving benefits for committing to extended subscription periods. This feature is designed to incentivize long-term engagement by offering tiered discounts based on the duration of the subscription plan. Users can choose from various subscription lengths, such as quarterly, semi-annual, or annual plans, and receive corresponding discounts. The feature ensures transparency by clearly displaying the discounted rates and savings compared to shorter-term plans. It aims to enhance user satisfaction and retention by providing financial flexibility and value.

### Acceptance Criteria

- Users must be able to view available subscription plans with their respective durations and discounted rates in a clear and organized format.
- The app should calculate and display the total savings for each long-term subscription plan compared to the monthly or shorter-term plans.
- Discounts must be applied automatically when users select a long-term subscription plan during the checkout process.
- The feature should support multiple payment methods, including credit cards, cryptocurrency, and other supported payment options, ensuring seamless transactions for discounted plans.
- Users must receive a confirmation of the discounted subscription purchase, including details of the plan duration, total cost, and savings, via in-app notification and email receipt.
- The app should provide a comparison chart or visual representation to help users understand the benefits of long-term subscriptions over shorter-term options.
- Discounts must be tiered based on subscription duration (e.g., 10% for quarterly, 20% for semi-annual, 30% for annual plans) and configurable by administrators for future adjustments.
- The feature must integrate with the billing system to ensure accurate invoicing and payment tracking for discounted subscriptions.
- Users should be able to upgrade or extend their current subscription to a longer-term plan and receive the applicable discount, with prorated adjustments if necessary.
- The app must include error handling to manage potential issues during the subscription purchase process, such as payment failures or network connectivity problems, and provide clear feedback to users.
- Discounts must be clearly communicated in marketing materials, subscription pages, and promotional campaigns to ensure users are aware of the benefits.
- The feature must comply with relevant financial regulations and tax requirements, ensuring accurate reporting and transparency in subscription pricing.
- Analytics should be integrated to track user adoption of long-term subscription plans and measure the impact of discounts on user retention and revenue growth.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and user experience.

## Customizable pricing plans for enterprise users

### Description

The 'Customizable pricing plans for enterprise users' feature in 'my-test-juan-0304' app allows Fund Managers to tailor subscription plans to meet the specific needs of their enterprise clients. This feature provides flexibility in pricing, feature allocation, and usage limits, enabling Fund Managers to offer bespoke solutions that align with their clients' trading requirements and organizational goals. It enhances the app's appeal to enterprise-level users by accommodating diverse business models and scaling requirements.

### Acceptance Criteria

- Fund Managers must be able to create and customize pricing plans directly within the app, including setting subscription fees, feature access levels, and usage limits.
- The customization interface should be intuitive and user-friendly, allowing Fund Managers to easily adjust parameters without requiring technical expertise.
- Pricing plans must support tiered structures, enabling Fund Managers to define multiple levels of access and features for different enterprise clients.

- The app should provide templates for common pricing models (e.g., flat rate, usage-based, tiered) to simplify the customization process.
- Fund Managers must be able to preview the customized pricing plans before finalizing them, ensuring accuracy and alignment with client expectations.
- Enterprise clients should receive clear and transparent breakdowns of their customized pricing plans, including detailed information on included features and associated costs.
- The app must support automated invoicing and payment processing for customized pricing plans, ensuring seamless billing and revenue collection.
- Fund Managers should be able to track the performance and usage of customized pricing plans, including metrics such as client retention, revenue generation, and feature utilization.
- The feature must include robust security measures to protect sensitive pricing and client data, adhering to industry standards for data encryption and access control.
- Customizable pricing plans must be compatible with the app's subscription management system, ensuring smooth integration and scalability.
- The app should provide analytics and AI-driven insights to help Fund Managers optimize their pricing plans based on client behavior and market trends.
- The feature must be tested across different devices and operating systems to ensure consistent functionality and user experience.
- The customization interface should adhere to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for Fund Managers with disabilities.

## Automated invoicing and payment reminders

### Description

The 'Automated invoicing and payment reminders' feature in the 'my-test-juan-0304' app streamlines the subscription management process by automatically generating invoices and sending timely payment reminders to users. This feature ensures users are informed about upcoming payments, overdue invoices, and subscription renewals, reducing the risk of service interruptions. It supports multiple payment methods, including cryptocurrency, and provides a transparent breakdown of fees. The feature is designed to enhance user convenience, improve financial compliance, and maintain uninterrupted access to the app's services.

### Acceptance Criteria

- Invoices must be automatically generated for all subscription tiers and sent to users via email and in-app notifications.
- Payment reminders must be sent at configurable intervals (e.g., 7 days before due date, on the due date, and after overdue) to ensure users are aware of upcoming or overdue payments.
- Invoices must include detailed information such as subscription tier, billing period, fee breakdown, applicable taxes, and payment methods available.
- Support for multiple payment methods, including credit/debit cards, bank transfers, and cryptocurrency, must be integrated into the invoicing system.
- Users must be able to view and download their invoices directly from the app in PDF format for record-keeping and compliance purposes.
- The system must provide notifications for successful payments and update the user's subscription status in real-time upon payment confirmation.
- Overdue payment reminders must include a clear call-to-action for users to make payments and avoid service interruptions, along with a grace period if applicable.
- The feature must support automated invoicing for enterprise users with customizable pricing plans and bulk invoicing options.
- Error handling mechanisms must be implemented to manage issues such as failed payments or incorrect invoice generation, with clear feedback provided to users.
- The invoicing system must comply with relevant financial regulations and standards, ensuring secure handling of payment data and user information.
- Analytics must be integrated to track payment patterns, overdue rates, and user behavior, providing insights for optimizing subscription management.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- The invoicing interface must be accessible, adhering to WCAG (Web Content Accessibility Guidelines) standards, ensuring usability for all users, including those with disabilities.

## Integration with financial software for subscription management

### Description

The 'Integration with financial software for subscription management' feature in 'my-test-juan-0304' app streamlines subscription management by enabling seamless integration with popular financial software tools. This feature allows Fund Managers to efficiently track, manage, and reconcile subscription payments, invoices, and billing details directly within their preferred financial software. By automating data synchronization and providing real-time updates, this feature reduces manual effort, enhances accuracy, and ensures compliance with financial reporting standards. It is designed to improve operational efficiency and provide a transparent view of subscription-related financial activities.

## Acceptance Criteria

- The app must support integration with widely-used financial software tools, such as QuickBooks, Xero, and FreshBooks, ensuring compatibility with industry standards.
- Fund Managers must be able to link their financial software accounts to the app securely, using OAuth or similar authentication protocols.
- Subscription payment data, including invoices, payment statuses, and transaction details, must be automatically synchronized with the connected financial software in real-time.
- The integration must support exporting subscription data in standard formats (e.g., CSV, Excel) for manual import into other financial tools if needed.
- Fund Managers must be able to view a summary of subscription-related financial activities within the app, including payment history, outstanding invoices, and subscription tier details.
- The feature must include error handling mechanisms to address issues such as failed synchronization, invalid credentials, or API rate limits, with clear feedback provided to the user.
- The integration must comply with relevant financial data security standards, including encryption of sensitive information during transmission and storage.
- Fund Managers must be able to configure notification preferences for subscription-related updates, such as payment reminders or invoice generation, within the app.
- The feature must provide a detailed audit log of all subscription-related financial activities, ensuring transparency and traceability for compliance purposes.
- The integration must be tested across different financial software platforms to ensure consistent functionality and performance.
- The user interface for managing financial software integration must be intuitive and accessible, with clear instructions for setup and troubleshooting.

## Transparent fee breakdown for users

### Description

The 'Transparent fee breakdown for users' feature in the 'my-test-juan-0304' app provides users with a clear and detailed view of all fees associated with their subscription plans and trading activities. This feature ensures that users can easily understand the cost structure, including subscription fees, performance-based fees, and any additional charges. By offering transparency, the feature builds trust and helps users make informed decisions about their subscription tiers and trading strategies. The fee breakdown is presented in an intuitive and user-friendly format, accessible from the user dashboard.

### Acceptance Criteria

- The fee breakdown must be accessible from the user dashboard under a dedicated 'Fees' or 'Billing' section.
- Users must be able to view a detailed breakdown of all fees, including subscription fees, performance-based fees, transaction fees, and any other applicable charges.
- The fee breakdown should include clear labels, descriptions, and amounts for each fee type, ensuring users can easily understand the cost structure.
- Performance-based fees must include calculations based on the high-water mark tracking system, with clear explanations of how the fees are derived.
- The feature must support multiple currencies, including cryptocurrency, and display fees in the user's preferred currency format.
- Users should be able to view historical fee data for previous billing cycles, allowing them to track their expenses over time.
- The fee breakdown must include visual aids, such as charts or graphs, to help users analyze their fee trends and patterns.
- The feature must provide a summary of total fees incurred for the current billing cycle, along with a projected estimate for the next cycle based on current activity.
- Ensure compatibility with the secure billing system, integrating seamlessly with payment methods and invoicing features.
- Include error handling to manage potential discrepancies in fee calculations or display, with clear feedback to users and a mechanism for reporting issues.
- The fee breakdown interface must be optimized for both desktop and mobile platforms, ensuring consistent usability across devices.
- Adhere to WCAG (Web Content Accessibility Guidelines) standards to ensure the fee breakdown is accessible to users with disabilities.
- Integrate analytics to track user interactions with the fee breakdown feature, providing insights for further improvements.

## AI-driven recommendations for optimal subscription tier

### Description

The 'AI-driven recommendations for optimal subscription tier' feature in the 'my-test-juan-0304' app leverages advanced AI algorithms to analyze user behavior, trading activity, and preferences to suggest the most suitable subscription tier. This feature ensures users maximize the value of their subscription by aligning their needs with the features offered in each tier. By providing personalized recommendations, the app enhances user satisfaction and helps users make informed decisions about their subscription plans.

### Acceptance Criteria

- The feature must analyze user activity, including trading frequency, portfolio size, and feature usage, to generate personalized subscription tier recommendations.



- Recommendations must be displayed prominently on the subscription management page, with clear explanations of why a specific tier is suggested.
- The AI algorithm must consider factors such as cost-effectiveness, access to advanced features, and user goals (e.g., active trading, passive investing, or fund management).
- Users must be able to view a detailed comparison of their current subscription tier versus the recommended tier, highlighting differences in features and pricing.
- The feature must provide actionable insights, such as how upgrading or downgrading tiers could impact their trading experience or cost savings.
- Recommendations must be updated dynamically based on changes in user behavior or market conditions, ensuring relevance and accuracy over time.
- The feature must include an option for users to provide feedback on the recommendations, which can be used to refine the AI algorithm further.
- Ensure the AI-driven recommendations are transparent, with users able to view the criteria and data points used to generate the suggestions.
- The feature must support all user roles (Active Trader, Passive Investor, Fund Manager) and tailor recommendations based on their specific needs and objectives.
- The recommendation system must comply with data privacy regulations, ensuring user data is securely handled and not exposed to unauthorized parties.
- Include error handling to manage scenarios where insufficient data is available to generate recommendations, providing users with alternative guidance or manual tier selection options.
- The feature must be compatible with the latest version of the app and tested across different devices and operating systems for consistency.
- Integrate analytics to track user engagement with the recommendations, such as the percentage of users who follow the suggestions, to measure the feature's effectiveness.

# Non-Functional Requirements

## Scalability and Performance

### Description

The application must handle increasing user loads and high-frequency data processing without degradation in performance.

### Acceptance Criteria

- Support up to 1 million concurrent users across all roles.
- Ensure response times for critical operations (e.g., trade execution, dashboard updates) are under 2 seconds.
- Maintain system uptime of 99.99% during peak trading hours.
- Implement dynamic scaling to handle market data volumes exceeding 10,000 updates per second.

## Cross-Platform Compatibility

### Description

The application must function seamlessly across multiple platforms, including web, iOS, and Android, to ensure accessibility for all users.

### Acceptance Criteria

- Provide full feature parity across web, iOS, and Android platforms.
- Ensure compatibility with the latest two major versions of iOS and Android.
- Support screen resolutions ranging from 720p to 4K.
- Ensure mobile app latency for critical operations is under 3 seconds.

## Security and Data Protection

### Description

The application must safeguard user data and transactions against unauthorized access and breaches, adhering to industry standards for security.

### Acceptance Criteria

- Encrypt all sensitive data at rest and in transit using AES-256.
- Implement multi-factor authentication for all user roles.
- Conduct quarterly third-party security audits with published results.
- Detect and respond to security threats within 1 second of occurrence.
- Ensure compliance with GDPR, CCPA, and relevant cryptocurrency regulations.

## High Availability and Disaster Recovery

### Description

The application must remain operational during unexpected failures and recover quickly to minimize downtime.

### Acceptance Criteria

- Achieve 99.99% uptime with failover mechanisms in place.
- Implement automated backups every 15 minutes.
- Ensure recovery time objective (RTO) is under 5 minutes.
- Ensure recovery point objective (RPO) is under 1 minute.
- Provide real-time monitoring and alerting for system health.

## Latency for AI Agent Operations

### Description

AI agents must process and respond to market data in real-time to ensure timely decision-making and trade execution.

### Acceptance Criteria

- Ensure AI agent decision-making latency is under 1 second for trade execution.
- Process market data updates within 500 milliseconds.
- Support inter-agent communication with latency under 200 milliseconds.
- Ensure feedback loop for refining agent decisions operates within 2 seconds.

## Accessibility Compliance

## Description

The application must be accessible to users with disabilities, adhering to recognized accessibility standards.

## Acceptance Criteria

- Comply with WCAG 2.1 Level AA standards.
- Provide text-to-speech functionality for visually impaired users.
- Ensure all interactive elements are operable via keyboard navigation.
- Include adjustable font sizes and color contrast settings.
- Test accessibility features with at least 100 users with disabilities.

## Integration Reliability

### Description

The application must ensure seamless integration with external exchanges and APIs, handling errors gracefully.

### Acceptance Criteria

- Support integration with at least 10 exchanges within 12 months.
- Ensure API response times for market data retrieval are under 2 seconds.
- Implement retry mechanisms for failed API calls with a maximum retry delay of 5 seconds.
- Log and alert errors for exchange-specific issues within 1 second of occurrence.

## Data Analytics Accuracy

### Description

The application must provide accurate and reliable analytics for trading performance and market insights.

### Acceptance Criteria

- Ensure analytics dashboards update within 2 seconds of new data availability.
- Achieve data accuracy of 99.9% for all performance metrics.
- Support historical data analysis for up to 10 years.
- Provide predictive analytics with a confidence level of at least 95%.

## Subscription Management Efficiency

### Description

The application must handle subscription and billing processes efficiently and securely for all user tiers.

### Acceptance Criteria

- Process subscription payments within 2 seconds.
- Support at least 10,000 concurrent subscription transactions.
- Ensure billing system uptime of 99.99%.
- Provide automated invoicing and payment reminders within 1 second of due date.

## User Experience Optimization

### Description

The application must deliver an intuitive and responsive user experience for all user roles and skill levels.

### Acceptance Criteria

- Ensure onboarding completion time for new users is under 5 minutes.
- Provide dashboard load times under 2 seconds.
- Support drag-and-drop customization with latency under 1 second.
- Ensure voice command recognition accuracy of 95%.
- Test user experience with at least 1,000 users across all roles.