

Product Requirements Document: Equity Quest Platform

Version: 3.1 (Final & Verified) **Date:** September 26, 2025 **Author:** Gemini AI Project Manager
Status: Approved for Development

1. Introduction & Overview

This document outlines the product and technical requirements for a web-based application to host "Equity Quest," a mock stock trading competition. This platform will serve as the central hub for all competition activities, managed through a powerful administrator control panel that manually drives all market events and price changes.

Project Goal: To build and deploy a robust, engaging, and institutional-grade application to host the competition, with a maximum development timeline of two weeks and no financial investment in software or infrastructure.

- **Core Philosophy:** The platform will not automate any price movements. Instead, it will empower an Organizer (Admin) with a real-time control panel to manually trigger every news event, price change, and market shock described in the competition rulebook.
- **Target Audience:**
 - **Participants (Teams):** Competing teams who need a seamless, real-time interface to trade, view their portfolio, and react to admin-pushed information.
 - **Organizers (Admins):** The competition runners who will actively manage the simulation's progression, acting as the "market makers" through a dedicated control panel.

2. Goals & Success Metrics

The primary goal is to flawlessly execute the competition, with an admin successfully directing all events from the source document.

Metric	Target	Measurement Method
System Uptime	100% during competition rounds	Server logs and uptime monitoring tools.
Core MVP Delivery	Base code for real-time updates functional by Sunday, Sep 28.	Successful test of admin price change reflecting on participant screen.
Feature Completeness	100% of required admin controls & participant features implemented	Checklist against this PRD.
Real-time Performance	Admin actions reflected on participant screens in < 2 seconds	Real-time monitoring of WebSocket and API latency.
On-Time Delivery	Application deployed and handed over within 14 days	Project completion date vs. start date.
Cost Compliance	\$0 infrastructure and software	Final bill of materials for all

Metric	Target	Measurement Method
	cost	services used.

3. Functional Requirements

3.1. Participant-Facing Application

- **FR-1: Real-time Dashboard:** A single-page application displaying an asset watchlist, the user's portfolio (value, cash, P&L), an order entry form, and a news feed. All data must update in real-time without a page refresh.
- **FR-2: Trading & Rules Engine:** The system must validate and execute all participant trade requests against the established rules:
 - **Starting Capital:** ₹5,00,000.
 - **Order Types:** Market, Limit, Stop-Loss.
 - **Constraint Enforcement:** All position limits (20% stock, 25% commodity, 40% sector), transaction costs (0.10%), and short selling margin requirements (25% initial, 15% maintenance) must be strictly enforced.
- **FR-3: Live Leaderboard:** A view that displays a real-time ranking of all teams based on their current portfolio value.
- **FR-4: Transaction History:** A view where participants can review a complete log of their past trades and orders.
- **FR-5: Information Display:**
 - A public "News Ticker" that displays headlines pushed from the Admin panel.
 - A private "Messages" inbox for receiving insider tips from the Admin.

3.2. Admin Control Panel

- **FR-6: Secure Admin Login:** A separate and secure login mechanism for the Admin to access the master control panel.
- **FR-7: Global Round Controls:** Buttons to START, PAUSE, and END each competition round. A master timer must be displayed.
- **FR-8: Price Manipulation Interface:** A dashboard with a list of all tradable assets and defined sectors. Next to each, there must be an input field where the admin can type a percentage (e.g., -5.5 or +8) and hit "Apply" to instantly change its price for all users.
- **FR-9: Information Dissemination Interface:**
 - A "Broadcast News" tool with a text box and a "Send" button to push public headlines.
 - A "Send Insider Tip" tool that allows the admin to select a team, choose the tip type, and send it as a private message.
- **FR-10: Event Macro Triggers:** A dedicated section with single-click buttons for each major pre-scripted event (e.g., "Trigger Telecom Shake-up," "Trigger IT Whiplash," "Trigger Black Swan"). These buttons will execute the multi-step impacts described in the rulebook automatically.
- **FR-11: Team Monitoring View:** A real-time table displaying the portfolio value, cash, rank, and detailed current holdings of all participating teams.
- **FR-12: Manage Information Trades:** A function for the admin to approve or reject a proposed "information trade" between two teams, which involves a 2% portfolio value transfer.

3.3. Scoring

- **FR-13: Automated Final Scoring:** Upon competition conclusion, the system must calculate and display a final ranked list based on the weighted formula: $\text{Score} = (\text{Portfolio Value} * 0.70) + (\text{Sortino Ratio} * 0.30)$.

4. Technical Architecture & Stack

(This section remains unchanged as the stack is still appropriate for the fully-defined features.)

Component	Technology/Service	Justification
Frontend	React.js + Vite with a UI Library (e.g., MUI)	React's component architecture and a pre-built library accelerate UI development. Vite provides an extremely fast development feedback loop.
Backend	Python with FastAPI framework	High-performance, native async support is crucial for handling many simultaneous WebSocket connections efficiently. Automatic data validation and API docs reduce development time.
Database	PostgreSQL	A robust, open-source relational database perfect for organizing users, portfolios, assets, and transaction logs with high data integrity.
Real-time Layer	WebSockets (via python-socketio & socket.io-client)	The most critical technical choice. A persistent, two-way connection is the only way to achieve the required sub-2-second performance for instant market updates pushed from the admin.
Hosting	Render.com (Free Tier)	A single, zero-cost platform that can host the PostgreSQL database, the Python/FastAPI backend, and the static React frontend site.
Code Repository	GitHub	Industry standard for version control, collaboration, and a single source of truth for the codebase.

4.1. Development Tools & Methodology

Tool	Role	Justification
Lovable / v0.dev	AI Frontend Scaffolding	Used to rapidly generate the initial React components and layouts for the Admin and Participant UIs directly from PRD feature descriptions, creating a visual baseline in hours, not days.
Cursor	AI-Powered Code Editor	Used as the primary IDE to accelerate backend development (API endpoints, DB models) and to help integrate and refine the AI-generated frontend components.

5. Execution Plan: The "MVP by Sunday" Sprint

(This section remains unchanged as the plan is designed to build the core first, then add these specific features.)

Phase 1: Foundation & MVP (Friday - Sunday)

- **Focus:** Establish the project, database schema, and the core WebSocket connection.
- **Goal:** An admin-triggered price change is reflected on the participant's screen in real-time.

Phase 2: Full Feature Development (Monday - Thursday)

- **Focus:** Build out the full trading logic and all participant/admin UI functionalities as defined in Section 3.
- **Goal:** All functional requirements are implemented and ready for testing.

Phase 3: Testing & Deployment (Friday - Sunday)

- **Focus:** Rigorous end-to-end testing of a full competition simulation.
- **Goal:** A bug-free, fully-functional application is deployed and handed over.

6. Risks & Mitigations

(This section remains unchanged.)

7. Out of Scope

(This section remains unchanged.)