

TradeJournal PRD

Product Requirements Document: TradeJournal MVP

- **Product Name:** TradeJournal (Working Title)
 - **Version:** 1.0
 - **Status:** Draft
 - **Date:** August 9, 2025
 - **Author:** Gemini & User
-

1. Introduction & Overview

TradeJournal is a web application designed for retail traders who want to improve their performance through meticulous record-keeping and data analysis. The platform allows users to log their trades, attach notes, and visualize key performance metrics. The goal is to provide traders with actionable insights into their trading habits, helping them identify winning patterns and eliminate costly mistakes. This document outlines the requirements for the initial release (MVP).

2. Problem Statement

Retail traders often struggle with emotional decision-making and a lack of objective performance data. Brokerage reports provide basic P&L but fail to answer the critical question: *why* a trader is winning or losing. Spreadsheets are a common alternative but are cumbersome to maintain, prone to errors, and lack the robust analytical and visualization tools needed to uncover meaningful patterns.

TradeJournal will solve this by providing a dedicated, user-friendly platform for trade logging and analysis, turning a trader's history into a source of valuable insight.

3. Target Audience

The primary user is the **Developing Retail Trader**.

- **Persona:** "Alex," a part-time trader with 6-18 months of experience. Alex understands basic market concepts but struggles with consistency and discipline. Alex wants to move

beyond just tracking P&L to understand which strategies, setups, or psychological states lead to the best results.

4. Goals & Objectives

- **User Goal:** To easily log every trade and identify actionable patterns to improve profitability.
- **Product Goal:** To provide a stable, intuitive platform for manual trade logging and basic performance analysis.
- **Business Goal:** To launch a functional MVP that proves the core concept and can be built upon with more advanced features.

5. Features & Scope (MVP)

The MVP will focus on the essential functionality needed to log and review trades.

Feature ID	Feature Name	Description	Priority
AUTH-01	User Authentication	Users must be able to sign up, log in, and log out. Trades are tied to a user's account and are private.	P0 (Must-Have)
TRADE-01	Manual Trade Entry	A form to manually log a new trade. Fields will include: Ticker Symbol, Entry Date/Time, Exit Date/Time, Time Zone, Direction (Long/Short), Entry Price, Exit Price, Quantity.	P0 (Must-Have)
TRADE-02	Automated P&L	The backend will automatically	P0 (Must-Have)

	Calculation	calculate the Gross P&L for each trade upon submission.	
TRADE-03	Trade Log View	A simple, chronological table view of all logged trades. It should display key data like Ticker, Direction, Quantity and P&L. Must be sortable by date.	P0 (Must-Have)
TRADE-04	Trade Detail & Notes	Clicking on a trade from the log will navigate to a detailed view showing all its data. This view will include a text-based journaling field where the user can add notes and reflections.	P0 (Must-Have)
DASH-01	Core Analytics Dash	A main dashboard view that displays high-level statistics calculated from all trades: Total P&L, Win Rate (%), Avg. Win (\$) and Avg. Loss (\$)	P0 (Must-Have)

Features Out of Scope for MVP (Future Versions):

- CSV / Broker statement import
- Advanced analytics (e.g., Sharpe Ratio, Profit Factor, Max Drawdown)
- Advanced filtering and tagging system (e.g., filter by strategy, time of day)

- Interactive P&L curve chart
- Screenshot attachments for journal entries

6. User Flow

1. A new user lands on the homepage and signs up for an account.
2. After logging in, they are directed to their empty Dashboard.
3. The user navigates to the "Log a Trade" page and fills out the form to submit their first trade.
4. They are redirected to the "Trade Log," where their new trade appears at the top. The Dashboard now reflects the data from this trade.
5. The user clicks the trade in the log to view its details and adds a note: "Followed my plan perfectly on this one."

7. Non-Functional Requirements

- **Technology Stack:**
 - **Frontend:** React
 - **Backend:** Python with FastAPI
 - **Database:** PostgreSQL
- **Performance:** All pages should load in under 3 seconds. Dashboard calculations for up to 1,000 trades should feel instantaneous.
- **Usability:** The interface must be clean, intuitive, and mobile-responsive for viewing and basic entry.
- **Security:** Passwords must be hashed. All communication between the frontend and backend must be over HTTPS.