Product Requirements Document: AI Trading Predictor

1. Executive Summary

Product Name: AI Trading Predictor

Status: MVP (Minimum Viable Product)

Objective: To provide retail traders with a machine-learning-driven tool that predicts short-term price movements for stocks and cryptocurrencies using technical analysis indicators.

---

2. Problem Statement

Retail investors often lack the technical expertise to combine multiple financial indicators (RSI, MACD, Moving Averages) into a cohesive price prediction. Additionally, many financial tools break when API access (like Yahoo Finance) is restricted, leaving users without a functional interface for testing or analysis.

---

3. Target Audience

- **Retail Traders:** Individuals looking for AI-backed second opinions on their trades.
- **Data Science Students:** Users interested in how XGBoost can be applied to financial time-series data.
- **Developers:** Those looking for a template for a full-stack ML application with robust fallback mechanisms.

---

4. Functional Requirements

4.1 Data Ingestion & Fallback

- **Primary Source:** Fetch real-time and historical OHLCV (Open, High, Low, Close, Volume) data via yfinance.
- **Fallback Mechanism:** If primary data is unavailable, the system must generate **synthetic data** using a random walk with drift to maintain app functionality.

4.2 Feature Engineering Engine

The system must automatically calculate the following for any given symbol:

- **Trend Indicators:** SMA (5, 10, 20, 50) and price-to-MA ratios.

- **Momentum:** RSI (14-day) and MACD (12, 26, 9).
- **Volatility:** Bollinger Bands and Standard Deviation of returns.

4.3 Machine Learning Model

- **Algorithm:** Utilize **XGBoost Regressor** for its ability to handle non-linear relationships.
- **Forecast Window:** Allow users to predict prices 1 to 30 days into the future.
- **Performance Metrics:** Display Mean Absolute Error (MAE) and a directional "Confidence" score.

4.4 User Interface (Front End)

- **Single-Asset Search:** Input field for tickers (e.g., TSLA, BTC-USD).
- **Batch Analysis:** Ability to analyze a list of symbols simultaneously and rank the best opportunities.
- **Visualizations:** Interactive multi-chart layout showing price, indicators, and feature importance.

---

5. Technical Stack

Component	**Technology**
Language	Python 3.x
Front End	Streamlit
Back End	Python (AITraderFixed Class)
ML Models	XGBoost, Scikit-learn
Charts	Plotly (Graph Objects & Express)
Data Handling	Pandas, NumPy

---

6. User Flow

1. **Input:** User enters a ticker and selects a forecast duration.
2. **Processing:** The Back End fetches data and generates features.
3. **Training:** The AI model trains on historical data and validates against a test set.

4. **Output:** The Front End displays a "Signal" (Buy/Sell/Hold) and interactive charts.

---

7. Future Enhancements (Phase 2)

- **Sentiment Analysis:** Integrate news and social media (Twitter/Reddit) sentiment into the feature set.
- **User Accounts:** Allow users to save a "Watchlist" and receive email alerts for "Strong Buy" signals.
- **Live Broker Integration:** Connect to APIs like Alpaca or Binance for automated paper trading.