Project Specification: IBKR Desktop AI Day Trading Agent (v1.6)

Goal: To develop a high-performance, real-time, AI-driven trading application on Replit that acts as an autonomous trader, utilizing the Interactive Brokers (IBKR) API for execution and the OpenRouter API for advanced, multi-model market analysis. **Crucially, the user must be able to perform all trading and account management tasks manually, independent of the AI agent.**

1. Overview and Technology Stack

Feature	Requirement	Notes
Platform	Replit (Python/Web Application)	Must be containerized and runnable within the Replit environment.
Trading API	Interactive Brokers (IBKR) API	Must connect via the TWS API or IB Gateway (preferably via an asynchronous Python wrapper like `ib_insync`) to ensure low-latency order execution and real-time data streaming.
AI API	OpenRouter API (Universal Access)	The system must be able to utilize **any model** available through the OpenRouter API simply by changing a configuration parameter (e.g., GPT-4, Claude, Gemini, etc.). This ensures maximum flexibility and allows for model comparison.
Language	Python (Backend Logic) & HTML/JS/Tailwind (Frontend UI)	Python is highly recommended for `ib_insync` and the AI/data processing logic.

2. Core AI Agent Logic (The "Trader")

The AI Agent must function as an autonomous, expert trader following a **Think-Analyze-Act (TAA)** loop, respecting the user's selected **Trading Horizon**.

2.1 Decision Parameters (User Inputs)

The agent must only execute trades based on user-defined parameters, specified via the UI:

1. **Profit Target (%):** The user-defined minimum expected percentage gain that justifies a trade. **The system must display an Agent Advisory Notification (2.1-5) suggesting an optimal target profit for the selected asset based on recent volatility and historical performance, which the user can accept or override.**
2. **Position Size ($):** The maximum dollar amount the agent can commit to a single trade.
3. **Position Size (Shares):** The maximum number of shares the agent can buy/sell/short.
4. **Risk Management:** Automatically implement a dynamic Stop-Loss (e.g., 2% below entry) and Take-Profit (e.g., 5% above entry) based on the user's overall risk profile.
5. **Agent Advisory Notification:** An LLM-generated recommendation for the Profit Target (2.1-1) for the current stock, derived from volatility, Average True Range (ATR), and current market conditions. This is a non-binding suggestion.

2.2 Execution Mandate and Order Types

The Agent must be capable of generating sophisticated order instructions that mirror the user's manual capabilities (Section 4.2), and these are then passed to the IBKR API for execution. **The Agent must be able to perform the same full range of trading tasks as the user.**

- **Full Order Parity:** The Agent must be able to generate and submit any order type available to the user, including **Limit, Market, Stop, Stop Limit** orders, and specify **Time in Force (TIF)** such as Day or GTC.
- **Actions:**
  - **Buy (Long):** Initiate a long position.
  - **Sell:** Close a long position or initiate a short-cover order.
  - **Short (Sell Short):** Initiate a short position.
  - **Cover (Buy to Cover):** Close a short position.
  - **Hold:** Take no action.

2.3 Trading Horizon Mandate

The Agent's behavior regarding holding positions must be strictly governed by the user's selection in the **Trading Horizon Control (Section 4.4)**:

- **Day Trading (Intraday):** The Agent targets closure within the trading day. However, if the **Profit Target (2.1-1)** is **NOT** met by market close, the Agent will **carry the position overnight** to the next trading day. The system must notify the user of any position carried overnight. Only the Stop-Loss will trigger an exit on the current day, regardless of time.
- **Positional Trading (Long Setup):** The Agent is allowed to hold positions overnight and longer, adhering only to the dynamic Stop-Loss and Take-Profit parameters set in Section 2.1. The TIF setting can default to GTC.

2.4 High-Frequency Prediction Engine (NEW)

The AI Agent must integrate a short-term predictive engine using the LLM and real-time data to determine immediate market direction. This analysis is mandatory for any trade signal generation.

- **Prediction Horizons:** The agent must generate distinct, confidence-weighted predictions for the next **1-minute, 5-minute, and 10-minute** intervals.
- **Prediction Output:** The output must be a determination of either a **Bullish Trend**, **Bearish Trend**, or **Consolidation/Neutral**.
- **Data Inputs for Prediction:** Analysis must primarily rely on Level 1 data, volume profile, short-term moving averages (e.g., 9/20 period), and real-time order flow changes, streamed from the IBKR API (Section 3).
- **Action Mandate:** Trades must only be executed if the prediction for at least the 5-minute horizon aligns with the intended direction (e.g., the trend must be confidently Bullish to execute a Buy order).

3. Real-Time Data Acquisition and Analysis (Low Lag Mandatory)

The agent’s intelligence is dependent on the speed and quality of its data. All data streams must be optimized for **minimal latency**.

Data Source	Requirement	Analysis Performed by Agent
IBKR Live Data	Real-time streaming market data (Level 1 and Level 2, if available via subscription) for all active and watchlist stocks.	Volume analysis, bid/ask spread monitoring, and current price comparison to historical data.
Watchlist Data	Continuous, real-time tracking of all user-selected stocks (contracts).	**Historical Analysis:** Comparison against 5-day and 30-day closing prices, 52-week highs/lows.
Market Sentiment	Integration with a reliable, low-latency financial news/sentiment API (e.g., Finnhub, Alpha Vantage) to track news, social media sentiment (Reddit, X/Twitter), and volume spikes.	Sentiment scoring (Positive/Neutral/Negative) and News impact assessment for selected stocks.
Trade Execution History	Real-time tracking of the user's portfolio, P&L, and margin usage via the IBKR API.	Profit target and Stop-Loss adherence monitoring.

4. User Interface and Interaction (Manual Trading Priority)

The UI must be designed to feel like a powerful desktop application (IBKR Desktop analogy) and provide all manual trading capabilities.

4.1 Chat Window (Agent Interaction)

A dedicated chat interface must allow the user to interact with the AI Agent using natural language for overrides, reporting, and configuration.

- **Query:** "Why did you buy 100 shares of AAPL 5 minutes ago?"
- **Override:** "Sell all my GOOGL shares now." (See Section 5.1)
- **Configuration:** "Only use 5% of my total cash for new trades."
- **Reporting:** "Summarize the current market sentiment for the technology sector."

4.2 Manual Trading Interface (Order Ticket)

The user must have a dedicated, intuitive panel to manually place orders, replicating a standard IBKR order ticket:

- **Symbol Lookup:** Field for entering/searching for stock ticker.
- **Action:** Select Buy, Sell, Sell Short, Buy to Cover.
- **Quantity:** Input field for number of shares.
- **Order Type:** Dropdown for Limit, Market, Stop, Stop Limit.
- **TIF (Time in Force):** Dropdown for Day, GTC (Good Til Canceled), etc.
- **Execution Button:** Clear buttons for **"Submit Buy Order"** and **"Submit Sell Order."**

4.3 Portfolio and Account Management

The UI must include real-time, streaming panels for core account management:

- **Portfolio/Positions Panel:** A real-time table showing all current holdings, including: Symbol, Quantity, Average Cost, Current Price, Market Value, Day's P&L, Total P&L, and Percentage Change.
- **Account Summary Panel:** Displaying key metrics: Total Equity, Available Cash, Buying Power, Maintenance Margin, and Excess Liquidity.
- **Orders Panel:** A panel to view all pending, executed, and cancelled orders with the ability to manually modify or cancel pending orders.

4.4 Control Panel and Safeguards

The UI must feature clear controls for the user to manage the agent's risk and autonomy.

- **Agent Execution Mode (Tickbox/Radio):** A crucial control allowing the user to set the AI's execution permission level. This setting dictates whether the Agent operates fully autonomously or requires user consent. Options must include:
  - **Full Autonomy (Execute):** The agent executes trades immediately based on its signals without requiring user approval (bypassing the Section 5.1 modal).
  - **Manual Approval (Modal):** (Default) The agent generates a signal, and the user must approve it via the Pre-Trade Override modal (Section 5.1) before execution.
  - **Observation Only (Disabled):** The agent continues to generate signals and analysis in the Chat Window, but no orders are sent to the IBKR API.
- **Trading Horizon Control (Radio/Toggle):** This control sets the strategic goal for the AI Agent (which governs Section 2.3).
  - **Day Trading**
  - **Positional Trading (Long Setup)**
- **Agent Status Indicator:** Clear visual indicator showing the currently active Agent Execution Mode and Trading Horizon.
- **Margin Control Toggle:** A toggle switch allowing the user to explicitly define:
  - `CASH ONLY` : Agent can only use settled cash for purchases. **(This applies only to AI-initiated trades.)**
  - `MARGIN ENABLED` : Agent can utilize available margin up to a user-defined limit (e.g., 50% of available margin).
- **Watchlist Management:** An intuitive interface to add, remove, and track stocks the agent should prioritize for analysis.

5. Safeguards and Stability (Critical)

5.1 Override Mechanism (User Control)

The user must be able to override the agent on all actions. The user's manual trades (from 4.2) always have immediate priority.

- **Pre-Trade Override (AI):** When the AI generates a `BUY` , `SELL` , or `SHORT` signal, and the **Agent Execution Mode** is set to **Manual Approval (Modal)**, the system must first present the user with a notification/modal detailing the proposed action (symbol, quantity, price target, AI reasoning). The

user must be able to click **"Approve"** (execute) or **"Reject"** (cancel) before the order is sent to IBKR.

- **In-Trade Override (AI/Manual):** The user must be able to manually close, adjust, or liquidate any position opened by the agent or manually via the chat interface or the dedicated trade management panel.

5.2 Error Handling and Resilience

- **API Fallbacks:** Utilize OpenRouter's model routing capabilities to ensure that if a primary LLM model fails or hits a rate limit, the request is instantly routed to a compatible fallback model, preventing agent downtime.
- **IBKR Connection Status:** Provide a clear, real-time status indicator of the IBKR API connection (Connected, Disconnected, Reconnecting).
- **Rate Limiting:** Implement exponential backoff and rate-limit handling for both the OpenRouter and IBKR APIs to prevent service denial.
- **Financial Constraint Check:** All trade actions must include a pre-flight check to ensure the trade adheres to the user's selected **CASH ONLY** or **MARGIN ENABLED** settings and does not violate IBKR's margin requirements.
- **Zero Lag Focus:** Given the real-time nature of the app, prioritize asynchronous programming