Key Points

1. **Flexible algorithm builder (not just if/then)**
    - Users pick indicators/settings, but instead of simple Boolean rules, each can have **weights**.
    - The app trains a **lightweight model** (think logistic regression or small NN) on those weighted settings → adapts to different market conditions.
    - Users can also **draw/upload trendlines** (or use community trendlines) and let the algorithm "learn the fit line" → bridging human intuition with automation.
    - **Virtual trading** (paper + backtesting) is critical to make this approachable — show simple outcome reports ("Sharpe ratio, max drawdown, win rate") so users see how their weighted model behaves.
2. **Tutorials + community library**
    - For beginners: explain *why* an indicator works (e.g., "RSI < 30 means oversold → historically bounce probability X%").
    - Provide **example starter algorithms** they can clone and tweak.
    - Community library of algorithms = strong value.
        - Yes: let creators choose **public vs private**.
            - *Public*: helps grow the ecosystem (like TradingView scripts).
            - *Private*: respects users who want to guard edge.
            - You can also allow **invite-only sharing** (small groups, mentorship).
3. **Trading personality analysis**
    - Challenge: users come here *to automate*, so you won't get much direct trade-history data from them.
    - Options:
        - **Questionnaire onboarding**: ask about drawdown tolerance, trade frequency, holding horizon, emotional comfort. (Similar to robo-advisor risk profiling.)
        - **Simulated choices**: give "Would you hold through this 20% dip for a bigger win?" scenarios.
        - **Behavioral data inside app**: how often they tweak, whether they prefer trend-following or reversal setups, whether they use tight stops or wide stops.
        - **Optional broker data sync** (with permission): pull their historical trades to better calibrate personality.
    - Output: classify them into archetypes (e.g., "Trend Rider", "Scalper", "Mean-Reversionist") and recommend algorithms/templates accordingly.

Suggestions

- **Explainability layer**: When the trained model makes a trade, show “Because indicator A had 70% weight, indicator B had 20%, trendline confirmed → opened long.” This builds trust and educates users.
- **Gamified learning**: Borrow from Duolingo-style progress → completing tutorials unlocks more complex algorithms.
- **Hybrid model**: Allow a spectrum → from strict rules (if/then) → to weighted ML model → to fully community-trained strategies. Users can progress along this path.
- **Safety guardrails**: Many retail traders blow up by over-optimizing. Build in warnings about overfitting and encourage walk-forward testing.
- **Personality + community blend**: Pair users with “mentors” or algorithm archetypes that fit their personality.

Summary (differentiator vs others)

- **Not just no-code** → it’s **flexible ML-enhanced models** with weights and trendline learning.
- **Not just copy trading** → it’s **guided education + explainability** with tutorials, example algos, and a community library (with public/private choice).
- **Not just rule-based risk scores** → it’s **trading personality profiling** to suggest algorithms suited to the user’s psychology.

This positions you closer to:

“The Duolingo + Spotify of trading algorithms: learn, personalize, simulate, and share.”