



Product Requirements Document (PRD)

Purpose: Design a semi-automated AI-driven trading bot using Zerodha's Kite Connect (with MCP) that maximizes net ROI (target 100%) on a ₹10,000 INR capital while minimizing trade count and costs. The bot supports both F&O (long calls/puts only) and cash equity trading, in intraday and swing modes, with strong risk controls and auditability.

Goals and Objectives

- **Primary Goal:** Maximize net return after all fees/slippage. *KPI:* Net profit % (after fees) over backtest and live periods.
- **Secondary Goals:** Limit drawdown (default $\leq 15\%$), protect capital (e.g. $\leq 5\%$ risk per trade ¹), ensure repeatable, auditable signals.
- **Operational Goals:** Robust backtesting/walk-forward testing; reliable live/paper trading via Kite MCP; clear monitoring/alerts; easy manual override (semi-auto mode).

Scope & Constraints

- **Broker/Platform:** Zerodha Kite Connect APIs (HTTP/WebSocket) and Kite MCP for AI integration.
- **Environment:** Python (3.10+) in VS Code. Use local or hosted MCP server (<https://mcp.kite.trade/mcp>) configured via VS Code Copilot Chat ².
- **Capital:** ₹10,000 INR starting (target 100% ROI as optimization target, not guaranteed).
- **Trade Fee/Slippage:** Estimate ~₹50–60 fixed per F&O trade. Model and minimize these costs in strategy logic.
- **Instrument Rules:**
 - **F&O:** Only *long* positions in call (CE) or put (PE) options; no naked short positions or selling. Strike selection uses OTM-to-ATM (delta 0.20–0.40) for leverage. Use nearest expiries (avoid last-day/week-before expiry for swing). Maintain margin and lot-size (NIFTY lot=75) constraints.
 - **Cash Equity:** Long-only stocks. Swing (delivery/NSE CNC) or intraday (MIS) trades allowed.
 - **General:** Both intraday and multi-day ("swing") strategies permitted in F&O and equities.

Success Metrics

- **Net Profit (after fees/slippage):** Primary metric over backtest and live/paper.
- **Trade Count:** Aim for low activity (configurable, e.g. ≤ 6 –12 swing trades/month; plus limited intraday triggers).
- **Average Cost per Trade:** Estimated and tracked (e.g. ₹50–60 F&O; display in logs).
- **Max Drawdown:** Configurable cap (default $\leq 15\%$) ¹; strategy aborted if breached.
- **Sharpe Ratio:** > 1.0 target (Sharpe > 1.0 is considered acceptable).
- **Win Rate & Profit Factor:** Trackable metrics; ensure robustness (profit factor > 1.5 as a heuristic).
- **Execution Slippage:** Median slippage $<$ configurable threshold (default $\sim 0.2\%$ of price). Model realistic slippage by volatility and execution delay.

- **Reproducibility:** All backtests fully reproducible; use walk-forward and MC tests for out-of-sample validation ³.

Acceptance Criteria

- **Backtest Performance:** Strategy yields positive net P&L after costs, with metrics meeting above targets, across 3+ years of historical data (in-sample and out-of-sample).
- **Trade Frequency:** Configured trade limits respected (no excessive intraday churn; >48h cooldown between new option positions by default).
- **Safety Limits:** System aborts new trades if drawdown, daily loss, or max concurrent positions thresholds exceeded ⁴.
- **Auditable Signals:** Each trade recommended with a “justification trace” (features→score→expected-return→decision). Logs capture full order lifecycle and P&L.
- **User Control:** In semi-auto mode, orders only sent upon explicit user confirmation. Fully-auto mode requires explicit consent toggle and kill-switch.
- **Technical Reliability:** No unhandled exceptions; network/API failures trigger pause and alerts.

System Architecture & Components

A **modular, multi-layered architecture** will separate responsibilities, ensuring low-latency data access, robust backtesting, and clear execution/risk controls:

• Data Layer

- **Market Data:** Historical and real-time candles (1m, 5m, 15m, 1d), tick/1s level trades, and (if available) L2 order-book snapshots for stocks and indices (NSE, NIFTY options). Data via Kite Connect (WebSocket) or vendor API.
- **Options Data:** Live option chain (strike, expiry, bid/ask, last price, implied IV, open interest) via Kite Connect or stored from periodic crawls. Include historical implied vol surfaces.
- **Auxiliary Data:** Corporate actions (splits, dividends), market holidays/calendar, margin requirements.
- **Storage:** Time-series optimized storage (e.g. Parquet files or TimescaleDB). Use columnar formats (e.g. Parquet) for fast reads/writes, given high data volumes. Ensure tick-level access for intraday backtests.

• Feature/Model Layer

- **Feature Store:** Compute and store indicators: price-based (EMA, RSI, ATR), momentum (multi-timeframe returns), mean-reversion signals (z-scores), volatility metrics, option Greeks (approx delta, vega proxies), implied vs realized vol comparisons, and microstructure features (order-book imbalance, depth). Also extract event features (earnings, FII data).
- **Machine Learning Models:** Ensemble of supervised models (e.g. XGBoost/CatBoost) that predict:
 - Probability of profit (classifier).

- Expected raw return (regressor).

These models use time-series-aware cross-validation and avoid lookahead. Optionally an RL layer for adaptive sizing. Use feature importance and simple rule models for interpretability.

- **Signal Scoring:** Combine ML outputs with rule-based filters (e.g. volume, delta bands) to score trade opportunities. Only high-confidence signals ($p \geq p_{\min}$, default 0.80) with expected net return above cost threshold are flagged ¹.

• Strategy Engine

- **Trade Filters:** Pre-filter candidates by liquidity (min OI/volume, tight spread), valid strike (delta 0.2–0.4 OTM by default), and IV conditions (e.g. IV spike or IV vs. historical). Exclude options in last day unless intraday scalping.
- **Signal Generation:** For each candidate, compute signal (direction, expected premium move, model score). Require $\text{expected_net_profit} \approx \text{predicted_gain} \times \text{premium} - (\text{fee} + \text{slippage_estimate}) \geq \text{threshold}$ (e.g. $\geq 2 \times \text{avg_cost}$, default $\sim \$150$) to proceed.
- **Position Sizing:** Use fixed-fractional or volatility-parity. Base risk on max_risk_pct (default 5%) of capital per trade ¹. For options, ensure total notional $\leq X\%$ of capital (default 30%). Respect minimum lot sizes. Adjust size so potential loss (entry-stop) \leq risk limit.

• Execution Engine

- **Order Manager:** Integrates with Kite Connect via MCP client. Implements order placement (limit, SL-M, market orders) and management. Default to limit orders near fair value; if unfilled in timeout_sec , optionally cancel or fallback to SL-M (for urgent intraday exits). Use MIS for intraday, NRML for delivery/swing. Bracket or OCO orders for automated TP/SL in intraday.
- **Retry & Reconciliation:** Monitor orders via Kite's APIs. On partial fills or failures, retry or cancel as per rules. Record actual fill prices and compute slippage vs expected. Update slippage model. All order actions and fills are logged.
- **Kite MCP Integration:** Authenticate via Kite Connect (token refresh logic). Use Kite MCP endpoints (`/order`, `/positions`, etc.) for trading and market data ². Ensure API rate limits and handle errors with exponential backoff and safe pause.

• Risk & Compliance Module

- **Pre-Trade Checks:** Reject orders exceeding per-trade or total exposure limits. Ensure sufficient margin (especially for options lots). Enforce max concurrent trades (default 3) and cool-downs between option entries (default 48h) to limit churn.
- **Stop-Loss & Time Exit:** For each position, enforce TP/SL (e.g. $\pm 40\text{--}100\%$ premium move or fixed INR) and time-based exit (close by EOD/MIS for intraday, >2d pre-expiry for swing options). Square off MIS daily as needed.
- **Circuit Breakers:** Automatic pause on severe drawdown, system error, or exchange halt. Include an emergency "kill switch" controlled by the user or triggered by safety rules ⁴.

UI / Manual Control

- **Order Preview Panel:** In VS Code or a lightweight web UI, display recommended trades (instrument, strike, expiry, side, lot, premium, p(win), expected P/L vs fee, suggested TP/SL). Allow one-click confirm or reject (semi-auto mode).
- **Modes:** *Semi-Auto (default)*: Bot suggests trades; user confirms. *Full-Auto (opt-in)*: Bot executes automatically with manual override/kill-switch. Always require explicit consent for live trading scopes.
- **Audit Trail:** Every recommendation, user action, and order event is logged with timestamps and rationale. Allow export to CSV/JSON.

Monitoring & Logging

- **Real-Time Metrics:** Track and display P&L, open positions, current drawdown, and latency. Plot equity curve and drawdown over time.
- **Alerts:** Send notifications for key events: trade execution, SL/TP hit, drawdown breach, API or system failure. Example alerts: "Daily loss >5% reached – trading paused."
- **Logging:** Detailed logs of signals, orders, fills, errors. Log average slippage per instrument for TCA. All logs persisted for auditing.

Algorithmic Strategy Design

A. General Entry/Exit Controls

- **High-Confidence Only:** Require model probability $\geq p_{\min}$ (default 0.80) and $expected\ net\ profit \geq safety_mult \times (fee + slippage)$ (default 2x, roughly $\geq \$150$ net after costs). This ensures trade value $\gg cost$.
- **Batching & Cooldown:** If multiple correlated signals appear together, select the top one to reduce trades. Enforce a cooldown between new F&O entries (default 48h) to avoid churn. Limit intraday triggers unless exceptionally high confidence.
- **Slippage/Aggregation:** Prefer large primary trades over multiple small ones to save on fixed fees. Always account expected slippage (e.g. from bid-ask spread) when calculating net P/L.

B. F&O Option Strategies

Swing Options (Multi-Day Holds)

- **Strike Selection:** Filter strikes by liquidity: require min open interest and volume. Prefer delta 0.20–0.40 (OTM-to-ATM) for directional plays to balance premium vs probability. ATM trades allowed if justified.
- **Expiry Choice:** Use nearest 1–2 weekly expiries. Avoid taking positions on expiry-day or the last day-of-week before expiry (unless a very short-term scalp). Target ~1–2 weeks duration to capture moves.
- **Position Sizing:** Cap total option exposure to $\leq 30\%$ of capital. If required premium for one lot $>$ allocation, either scale down to smaller position or skip. Ensure margin availability.
- **Filters:**

- **Implied Vol:** Enter if IV spike suggests directional move OR if IV is historically low and mean-reversion is expected. (Use IV/HV spread indicator to time trades).
- **Liquidity/Spread:** Exclude options with bid-ask spreads above threshold (e.g. >5% of premium) to minimize cost.
- **Exit Rules:**
- **Profit Target:** Close when premium rises by a set percentage (e.g. +40–100%) or fixed INR profit that comfortably exceeds cost.
- **Stop Loss:** Strict, e.g. stop if premium drops by -40% or more of entry cost, or a fixed INR loss limit. Implement via SL orders.
- **Time Exit:** If neither TP nor SL hit, close option trades a few days before expiry (e.g. 2 days prior) or on end-of-day for MIS. Prevent EOD carrying of weekly F&O to next cycle.

Intraday Options (Scalps)

- **Setup:** Use only very high-confidence, short-term signals (e.g. order-flow or sharp momentum). Aim for small net INR gains much larger than cost.
- **Orders:** Use limit or SL-M. If not filled quickly, cancel. Use bracket orders (if supported) for automatic TP/SL. Always square off at EOD via MIS.

C. Cash Equity Strategies

Swing Trades

- **Selection:** Identify high-liquidity stocks passing a momentum or mean-reversion screen. Avoid microcaps to keep spreads low.
- **Position Sizing:** Max 10% of capital per stock (configurable). Use CNC and hold days-weeks.
- **Timing:** Enter on multi-day signals to reduce churn. Use technical indicators (EMA cross, breakout, etc.) as rules for entry/exit.
- **Exit:** Use target R/R or volatility-based stops (e.g. 2×ATR).

Intraday Trades

- **Selection:** Only very clear setups (e.g. opening gaps, strong reversals). Use MIS bracket orders.
- **Risk:** Keep trade count low; emphasize swing over scalping to reduce fees.

D. Portfolio Allocation

- **Diversification:** Allow concurrent positions across F&O and equities, but cap total exposure. E.g. $\leq 30\%$ in options, $\leq 50\%$ in equities, leaving cash buffer.
- **Rebalancing:** Do not exceed max concurrent trades (default 3).
- **Reinvestment:** Profits accumulate as capital for new trades.

Data Specification

- **Historical Data:** Acquire 3+ years of minute-level (1m, 5m, 15m) and daily data for NIFTY, Bank NIFTY, and key liquid stocks. Include pre-market if needed.

- **Tick/Order-Book:** If available, log tick data and L2 quotes for simulating execution. Otherwise, build slippage model from minute data spread.
- **Option Chain:** Historical option chains (all strikes/expiries daily) including OI and IV.
- **Corporate/Calendar Data:** Corporate actions, earnings dates (for underlying stocks), NSE holidays. Fetch from public sources or APIs.
- **Data Storage:** Use a columnar time-series store (Parquet files or a TSDB) for fast slice access. Ensure historical data can be replayed for backtesting.

Feature Engineering & Model Pipeline

- **Features:** Calculate indicators on historical data:
- **Trend/Momentum:** EMAs, RSI, MACD, price returns over multiple windows.
- **Mean-Reversion:** Z-scores of recent price relative to moving average.
- **Volatility:** ATR, rolling std-dev, VIX/India VIX if relevant. For options: implied IV vs historical vol spread.
- **Order-Book (if used):** Bid-ask imbalance, trade volume spikes, time-in-force surges.
- **Option Greeks:** Approximate delta, vega (via option pricing or proxies). Delta used in strike filter.
- **Macro/Events:** (Optional) Index trend, news sentiment flags.
- **Model Pipeline:**
- **Training:** Use rolling windows for ML: train on last N (e.g. 3y) to predict next period. Use time-series CV (no lookahead).
- **Prediction:** For each new day/minute, compute features and feed into models to score trade candidates.
- **Calibration:** Regularly update/validate model (weekly/monthly retraining).
- **Outputs:** Each candidate gets (prob_win, expected_return). These feed into filter logic (e.g., only if prob > `p_min` and expected_return > cost threshold).

Position Sizing & Money Management

- **Risk per Trade:** Default 5% of current capital (configurable) `1`. Compute trade size so that $\text{entry_stop_risk} \approx \text{max_risk}$. For stocks, this is share count; for options, number of lots.
- **Position Limits:** Max cap on any single instrument (e.g. 30% of capital). Prevent cornering.
- **Leverage:** Only equity delivery (CNC) is unleveraged; MIS used intraday with standard margin but ensure worst-case losing scenario is within risk limit.
- **Fees/Slippage Buffer:** When sizing, account for fixed costs: effective additional risk_per_trade = risk_per_trade + (fee+slippage).

Execution Logic (Kite MCP Integration)

- **Authentication:** Use Kite Connect OAuth. Manage access/refresh tokens. Ensure user explicitly grants trading scopes.
- **Order Placement:**
- **Default to Limit:** Place limit buy orders at current ask (options/stocks) or bid price with small spread allowance.

- **Fallbacks:** If limit not filled in `limit_timeout` seconds (e.g., 5s), cancel to avoid adverse market moves. For urgent exits, use SL-M or market to ensure execution.
- **Brackets:** Use Coil/BO orders (if enabled) for TP/SL on intraday legs. Note: Kite Connect supports MIS, NRML and CO/BO orders.
- **Reconciling Fills:** After each order, poll Kite to confirm fill status. If partially filled, user may adjust or cancel remainder. Update actual cost and compare to expected slippage.
- **Rate Limits & Retries:** Respect Kite API rate limits. On failure (network/timeout), retry with backoff or pause trading after repeated failures.

Testing & Validation Plan

- **Backtesting Engine:** Develop a vectorized backtester that:
- Applies exact signal and execution logic on historical data.
- Simulates transaction costs: fixed fees, taxes, and slippage (modeled by volatility/spread).
- Enforces lot-size and position sizing rules.
- **Walk-Forward Testing:** Use rolling-window walk-forward (e.g. train on 5y, test next 1y, slide forward). Aggregate out-of-sample results for realistic performance.
- **Monte Carlo Robustness:** Perform Monte Carlo robustness tests (reshuffle trades or returns) to gauge variability ³. Ensure strategy outperforms random/reshuffled baselines.
- **Paper Trading:** After backtests are satisfactory, run 30–90 days of live-simulated trading via Kite MCP (paper mode), comparing signals to fills. Monitor drift between expected and actual performance.
- **Stress/Edge Cases:** Test scenarios: data gaps, false signals, rapid price moves (e.g. circuit breaker hits), API downtime. Verify system pauses/triggers alerts correctly.
- **Reporting:** Generate reports summarizing backtest results: equity curve, drawdowns, trade list, per-trade P/L breakdown (including costs) to justify decision-making.

Monitoring, Logging & Alerting

- **P&L Tracking:** Real-time display of current unrealized P/L, cumulative net P/L, and running Sharpe.
- **Risk Metrics:** Live drawdown calculation, margin usage, and position exposure gauges.
- **Latency & Performance:** Log time-to-decision and time-to-order for each signal to detect slowdowns.
- **Alerts:** Configurable alerts (e.g. email/SMS) for conditions such as: daily loss exceeds X%, order rejection, service/API down, high network latency, or reaching max positions.
- **Audit Logs:** Persist all signals, trades, model versions, and user actions. Example: "Trade opened: BankNIFTY CE 45,000 on 2025-01-15 @ ₹120 (p=0.85, exp_PL=+₹500 vs cost=₹60)".

Safety, Compliance & Consent

- **User Consent:** Explicit consent required for live trading. Show clear disclaimers ("No guaranteed returns; trading risks apply").
- **OAuth Scopes:** Request only necessary Kite MCP scopes. In UI, separate read-only analytics from trade-write capabilities.
- **Kill-Switch:** Implement an emergency stop to halt all trading (e.g. on critical bug or user panic).

- **Regulatory:** Adhere to Zerodha's API usage policies and NSE regulations. For example, do not exceed order limits or manipulate markets.
- **Transparency:** Provide user with summary of strategy rules, so they know what the bot does.

Implementation Roadmap & MVP (30-60 days)

• **Week 0-2:**

- Set up data ingestion (download/store historical candles and options).
- Build basic backtest engine. Validate transaction cost model (ensure ~₹50-60 F&O fee impact matches expectation).

• **Week 2-4:**

- Implement rule-based option-only strategy (entry/exit filters, sizing, P/L tracking). Backtest to verify profitability and low trade count. Adjust thresholds to meet metrics.

• **Week 4-6:**

- Add ML components (train classifier/regressor on features). Replace or augment rule filters. Perform walk-forward and Monte Carlo tests ³.

• **Week 6-8:**

- Integrate Kite MCP in VS Code. Develop UI for order preview & confirmation. Test order placement on Kite's sandbox/paper mode.

• **Week 8-12:**

- Hardening: robust logging, alerting, error handling. Test full end-to-end on live data (paper). Prepare user documentation for setup/consent. Optionally enable limited auto-mode with safeguards.

Algorithms & Pseudocode

Below are representative pseudocode outlines for key components. (All calculations use current capital and cost parameters from config.)

```
# Strike Selection (F&O) Pseudocode
for option in option_chain_today:
    if option.type in ['CE', 'PE']:
        if avg_open_interest(option) >= min_OI and avg_volume(option) >=
min_vol:
            delta = option.delta_estimate
            if delta_min <= abs(delta) <= delta_max:
                if (option.bid / option.ask - 1) * 100 <= max_spread_pct:
                    candidates.append(option)
# Choose best candidate by expected net return (after cost) or probability.
best_option = argmax(candidates, key=lambda opt: model_expected_return(opt) -
estimated_cost(opt))
if best_option:
    evaluate_trade(best_option)
```

```

# Expected Return Calculation
predicted_gain = model.predict_gain(option) # % premium increase
expected_profit_inr = predicted_gain * option.premium - (fee +
slippage_estimate)
if expected_profit_inr >= min_expected_net_profit:
    signal_confident = (model.predict_prob(option) >= p_min)
    if signal_confident:
        ready_to_trade = True

```

```

# Position Sizing
risk_per_trade = max_risk_pct * current_capital
trade_risk = (entry_price - stop_loss_price) * contracts * lot_size
if trade_risk > risk_per_trade:
    contracts = floor(risk_per_trade / ((entry_price - stop_loss_price) *
lot_size))
# Ensure at least 1 lot if possible; otherwise skip trade.

```

```

# Order Placement & Reconciliation
if ready_to_trade:
    order_id = kite.place_order(
        tradingsymbol=instrument,
        exchange="NSE",
        transaction_type="BUY",
        quantity=contracts*lot_size,
        order_type="LIMIT",
        price=limit_price,
        product="MIS" or "NRML"
    )
    wait_until_executed(order_id, timeout=timeout_sec)
    status = kite.order_status(order_id)
    if status == 'COMPLETE':
        fill_price = kite.get_fill_price(order_id)
        actual_slippage = fill_price - expected_fill_price
        log_order_execution(order_id, fill_price, actual_slippage)
    else:
        kite.cancel_order(order_id)
        log("Order cancelled or failed", order_id, status)

```

```

# Order Reconciliation (periodic check)
for order in open_orders:
    status = kite.order_status(order.id)
    if status in ['COMPLETE', 'CANCELLED', 'REJECTED']:
        reconcile(order)

```

```
        elif status == 'PENDING' and elapsed_time > max_wait:  
            kite.cancel_order(order.id)  
            reconcile(order, cancelled=True)
```

Configuration Template (YAML)

```
capital_initial: 10000 # INR starting capital  
p_min: 0.80          # Min probability threshold for entries  
min_expected_net_profit: 150 # INR (2x avg cost)  
avg_fee_per_trade: 60   # INR (F&O trade fee/slippage)  
max_risk_pct: 0.05    # Max 5% of capital per trade  
option_max_exposure_pct: 0.30 # 30% of capital in options  
delta_min: 0.20  
delta_max: 0.40  
min_option_OI: 5000    # example threshold  
max_bid_ask_spread_pct: 5 # %  
cooldown_hours: 48      # Cooldown between new option trades  
drawdown_limit_pct: 0.15  
daily_loss_limit_pct: 0.05  
max_concurrent_trades: 3  
slippage_model: "volatility_based" # or 'fixed', 'historical_spread'
```

Internal References: This design follows industry best practices for algorithmic trading architecture and risk management ^①, and incorporates Kite Connect/MCP specifics as per Zerodha documentation ^②. All strategies and rules are parameterized per the above config for easy tuning.

^① ^④ Risk Management Strategies for Algo Trading

<https://www.luxalgo.com/blog/risk-management-strategies-for-algo-trading/>

^② Connect your Zerodha account to AI assistants with Kite MCP – Z-Connect by Zerodha

<https://zerodha.com/z-connect/featured/connect-your-zerodha-account-to-ai-assistants-with-kite-mcp>

^③ Robustness Tests and Checks for Algorithmic Trading Strategies | Complete Guide – Build Alpha

<https://www.buildalpha.com/robustness-testing-guide/>