Prediction Pro Simulator Whitepaper – Condensed Prototype Specification for bolt.new

Version 1.4

Date: November 12, 2025
Author: Aleksi Grym (Founder, based on discussions with Grok AI)
Project Name: Polymarket IQ - Prediction Pro Simulator
Tagline: Simulate to Dominate

Summary for bolt.new Prototype Building

This whitepaper is a condensed and enhanced version of the previous document, designed specifically for use with the bolt.new tool to create a prototype. It focuses on the core information for the MVP (Minimum Viable Product): one simulated prediction market ("Will Trump sue the BBC?"), the starting order book (based on the attached image), the design model (based on the attached Polymarket design images), the simulation flow (now with 3 alternative event paths, each with 5 unique news events and price changes defined in the attached "Market and News Content Spec v1.0ent.pdf"), the user's decision-making after each event (with multiple actions, side switching, precise amounts, and action removal), the outcome (varying Yes/No resolutions across paths), and feedback (multiple varied options with comprehensive end-report including graphics).
Everything is explained in great detail so that bolt.new understands exactly what to build: A clickable web-based prototype where the user can experience the entire simulation from start to finish, with randomization for replayability. The prototype does not need to be a perfect production version – it is sufficient that it is an interactive demo with one market, using no-code/low-code elements (e.g., React-based UI, simple backend logic for events). Integrate the attached images as design references: The order book image for the starting state, the Polymarket design images for the general UI model (dark theme, cards, navigation), the market header profile image for the market card, breakingnews.png for news headers, logo.gif for the Polymarket IQ logo, and xlogo.png for the X link.
Goal: Create a prototype where the user can start the simulation (with initial loading), react to news in real-time with multiple trades (affecting order book), see price changes, order book updates (dynamic and living), comments, request AI analysis, and receive a final comprehensive

report with history tracking. This helps test the concept without full development, ensuring accurate PNL calculations, mirror pricing, budget limits, and smooth transitions.
For news events, comments, price impacts, and resolutions: Do not duplicate or generate dynamically; refer exclusively to the attached "Market and News Content Spec v1.0ent.pdf" for all hardcoded content (market rules, 15 unique news articles/summaries, comments per event, etc.). Implement as static data arrays from that document.

---

1. Introduction and Prototype Objectives

1.1 Overview

Prediction Pro Simulator is a web-based simulator that mimics prediction market trading (such as Polymarket) in a risk-free environment. The prototype focuses on one market as an MVP: The user starts with $1000 simulated USD, opens a position (with options to hold both Yes and No simultaneously), the simulation progresses with accelerated time through one of 3 randomized alternative event paths (each with 5 unique news events changing the price, as defined in "Market and News Content Spec v1.0ent.pdf"), the user makes multiple decisions after each event (with action chaining, removal, and "Done" confirmation), the simulation ends with a varying resolution (Yes or No based on path), and provides varied feedback in a comprehensive report with graphics. Upon replay, randomly select a different path for variety.
For bolt.new: Build this as a simple web application. Use the attached design images as a model: Polymarket's dark theme (dark blue background #001133, white text, rounded cards with percentages and Buy Yes/No buttons). The simulation UI should resemble Polymarket but simplified – show the order book (switchable between Yes/No views), position management (tracking both sides with budget checks), news notifications as popups or on a timeline (with hide only, no close), a chat-like comments section on the market page, an AI analysis button, and a history box for past simulations.

1.2 Prototype Scope

- **One market:** "Will Trump sue the BBC?" (details below).
- **Starting state:** Order book exactly as in the attached image, with switchable Yes/No views.
- **Simulation duration:** Randomized selection of 1 out of 3 alternative paths (from "Market and News Content Spec v1.0ent.pdf"), each with 5 news events, user decisions after each (multiple actions allowed).
- **End:** Resolution varies by path (e.g., 2 paths end in No, 1 in Yes, per spec PDF), feedback based on user's positions and actions in a detailed report.
- **Additional features:** Comments chat box (updated per event from spec PDF), AI analysis button (requestable at any of 5 stages), history box for all simulations, dynamic

order book living updates, budget limit ($1000 USD), action removal, loading windows (including start), news hide only, news header images, logos with links.
- **No required integrations:** Static data (no APIs, no external AI yet – hardcoded analyses and data from spec PDF). Use JavaScript for price calculations (ensuring mirror logic and accurate PNL), state management, and randomization (e.g., Math.random() to pick path on replay).

---

2. Problem Definition (Brief Context)

New prediction market traders fear losing real money. The prototype solves this by offering gamified training: The user learns to react to news with multiple trades (affecting order book), manage dual positions within budget, analyze comments and AI insights, and track history without risks, with realistic price mirroring and PNL.

---

3. Solution: Detailed Description of the Prototype Simulation

3.1 Example Market: "Will Trump sue the BBC?"

This is the prototype's only simulated market. The user sees it as a Polymarket-style card (based on the attached design images: Title, image of Trump, Yes/No percentages, Buy buttons).
Market Header: Include a market profile image (attached as a separate file for bolt.new – use it as the header image for the market card, showing Trump or BBC-related visuals to make it visually engaging, placed at the top of the market page with the title below it).
Market Description (Display in the prototype as a popup or on the page): Refer to "Market and News Content Spec v1.0ent.pdf" for full rules (resolution, qualifications, sources).
Starting Order Book (Based on the attached image):

- Display the order book exactly as in the image at the start of the prototype. Ensure YES and NO prices are always mirrors: If YES is X¢ (X%), NO must be (100 - X)¢ ((100 - X)%) – prices sum to 100¢ (100%) excluding spread. Initially, show Yes order book (as in attached image), but allow switching: User clicks "Yes" or "No" tab/button above the order book to toggle views. When "No" is selected, display a mirrored order book (e.g., asks and bids inverted: a 73¢ Yes ask becomes a 27¢ No bid, with shares and totals recalculated accordingly to maintain realism – for example, total for inverted bid = shares * (1 - price/100)).
    - **Yes View (default, as attached):**
        - Asks (Sell offers, red shade):
            - 73¢: 520.00 shares, Total $1344.19
            - 72¢: 499.45 shares, Total $964.59

        - 71¢: 831.43 shares, Total $604.99
        - 70¢: 20.96 shares, Total $14.67
    - Bids (Buy offers, green shade):
        - 69¢: 571.16 shares, Total $394.10
        - 68¢: 509.87 shares, Total $740.81
        - 67¢: 642.00 shares, Total $170.95
        - 65¢: 30.00 shares, Total $190.45
        - 63¢: 500.00 shares, Total $1505.45
    - Other info: Last 68¢, Spread 1¢.
- **No View (mirrored):** Invert logic – e.g., Yes 73¢ ask = No 27¢ bid, with shares and totals adjusted proportionally (calculate as 100 - price, but ensure visual symmetry; for total, use shares * (1 - price/100) to mirror value).

- The user starts with $1000 simulated USD budget. When attempting to buy more than available funds, display an error notification (e.g., popup: "Insufficient funds. You have $X remaining. Adjust amount or sell positions.") and prevent the trade. Track budget in real-time: Deduct on buys, add on sells; if budget < trade amount, block execution.

Comments Section: Refer to "Market and News Content Spec v1.0ent.pdf" for all hardcoded comments (5 per event, with usernames, mix of value/troll). Add a comments field on the market page (below the order book, not in the simulation window), styled as a chat box (scrollable list with user avatars/icons, timestamps, and text bubbles for realism, like a Discord or forum thread). It updates with each news event: Add the 5 new comments from the spec PDF tied to that event/path.

Portfolio Management and PNL Calculation (Detailed Logic): Track user's positions separately for Yes and No (allow holding both simultaneously). Use precise calculations to avoid errors:

- For each buy: Update average buy price using weighted average: new_avg = ((old_shares * old_avg) + (new_shares * buy_price)) / (old_shares + new_shares). Deduct from $1000 budget.
- For each sell: Realize PNL = (sell_price - avg_buy) * sold_shares (positive if profit, negative if loss). Add proceeds to budget.
- Unrealized PNL for held positions: (current_price - avg_buy) * shares (must be negative if current < avg_buy, e.g., buy No at 28¢, current 20¢: PNL = (20 - 28) * shares = -8¢/share * shares = negative total).
- Display in portfolio summary box (e.g., "Yes: X shares @ avg Y¢, unrealized PNL +$Z (current W¢) | No: A shares @ avg B¢, unrealized PNL -$C (current D¢) | Realized PNL: $E | Total PNL: $F | Budget: $G / $1000"). Aggregate total PNL = unrealized Yes + unrealized No + realized. Round to 2 decimals, use floating-point. Example fix: If YES position shows +$182099.25 for 9391.49 shares @ avg 68.61¢, current 88¢ – verify: PNL = (88 - 68.61) * 9391.49 ≈ 19.39 * 9391.49 ≈ $182,099 (correct if inputs match; if bugged, ensure no positive on drops by strict < comparison in code).

AI Analysis Button: On the market page, add a button "Request Polymarket IQ AI Analysis" (styled as a premium feature, e.g., blue button). User can request it at any of the 5 stages (before/after each news). Each request generates a unique hardcoded analysis popup: Summary of current market state, Monte Carlo simulation (e.g., "Based on 1000 sims, Yes probability 62% with avg payout $1.20"), AI-"found" news (1-2 extra fictional articles with summaries), and

decision tips. Vary by stage/path (hardcode 15 variations: 5 stages x 3 paths; refer to spec PDF if expanded).

History Box: Add a persistent "Simulation History" box on the market page (e.g., sidebar or tab), showing all past simulations: List each (e.g., "Simulation 1: Path A, Resolved Yes, PNL +$47"), with details (positions, actions). Summary at top: Total volume traded ($ sum), net PNL (sum wins/losses), number of simulations. Persist via local storage for replay sessions.

Logos and Links:

- Integrate logo.gif (attached) as the main Polymarket IQ logo: Place in header/top-nav, sized 150x50px, animated (ensure loop plays smoothly as GIF).
- Integrate xlogo.png (attached) as X (formerly Twitter) icon: Place in footer or nav, link to "https://x.com/PolymarketIQ" (click opens new tab).

3.2 Simulation Flow: Step-by-Step in Detail

The prototype simulates time in an accelerated manner (e.g., timeline slider or automatic progression). Randomly select 1 of 3 alternative paths at start/replay (e.g., if(Math.random() < 0.33) path1 else if <0.67 path2 else path3). Each path has 5 unique news events, changing prices realistically (escalation raises Yes, de-escalation lowers), maintaining YES + NO = 100¢. 1 path resolves Yes, 2 resolve No for balance. Refer to "Market and News Content Spec v1.0ent.pdf" for all hardcoded news (summaries, full articles, impacts, comments, resolutions).

Step 1: Simulation Start

- User selects position (Yes/No) and amount (default $100, max based on $1000 budget).
- Show current position (e.g., "You hold 147 Yes shares at avg 68¢").
- Start "fast-forward" (e.g., button "Start Simulation") – immediately show 4-second loading window ("Simulating initial market...") before first news.

Step 2: News Events and Price Changes

- Each news appears sequentially (automatically or with a "Next Event" button).
- News is displayed as a summary in the notification (modal popup), with attached breakingnews.png as header image (top of popup, sized to fit, e.g., 300x100px, for authentic news feel). Clickable to open full article in expanded view.
- Simultaneously with news display: Update order book to reflect new price (e.g., if YES rises to 75¢, shift all asks/bids accordingly – lower levels fill in randomly to maintain depth, e.g., add new bid at 74¢ with 100 shares if old top removed; logic: Recalculate entire book by shifting entries up/down by change amount, preserving spread/volume ratios for realism – specifically, for each level, new_price = old_price + delta if rising, adjust shares/totals proportionally to simulate market reaction; ensure mirror on No view).
- After each news, show a 4-second loading window (e.g., spinner with "Simulating market reaction...") for a smoother transition, not abrupt.
- User can perform multiple actions: First choose YES/NO side (allow switching sides), then BUY/SELL, then amount (All or specific sum, e.g., input $50). User can chain actions (e.g., Sell all Yes, then Buy No $100) using "Action" button to execute each (add to pending list), with option to remove any pending action (e.g., trash icon next to each in a list view) before pressing "Done - Continue simulation" to commit and proceed. If action exceeds budget, block and notify.

- User's trades affect order book: If user buys the lowest ask (e.g., 70¢ for 100 shares Yes), remove that bundle entirely; shift next ask down to become new lowest (e.g., 71¢ becomes 70¢ if no fill). For sells, add to bids. Ensure mirror update on No side. Additionally, make order book "live": Every 2 seconds, alternate Yes/No side: Randomly remove a random bundle (e.g., mid-level bid) and add a new one at similar level (e.g., +1-2¢ variance, 50-200 shares) without changing market price – simulate liquidity flow (use JS timer for updates, e.g., setInterval(updateOrderBookRandomly, 2000)).
- If the user switches sides (e.g., sell Yes and buy No), allow it, but calculate profits/losses in real-time.
- The "breaking event" window (news notification) is always active but only hideable (minimize button, no X/close to prevent accidental loss – hidden window can be re-opened via "Show News" button on market page) so the user can check price development and order book mid-simulation without it blocking the view.
- Update comments section with each event: Add the 5 new comments from the spec PDF tied to that event/path.

Step 3: Simulation End and Resolution

- After all 5 news: The market resolves based on path (show notification accordingly, from spec PDF).
- Calculate user's final outcome: If held Yes on Yes resolve or No on No, profit ($1 per share for winner). Aggregate PNL accurately.
- Show summary: Profits/losses ($ amount), position history (what was done at each step).

Step 4: Feedback (Positive/Negative)

- Show comprehensive "Simulation Complete and Performance Summary" report page: Include graphical elements for engaging experience (e.g., PNL line chart over 5 events, position pie chart Yes/No, bar graph of actions per stage – use Chart.js or similar for visuals). Polymarket IQ AI text analysis: General summary (e.g., "You performed well by holding through dips, netting +47%"), plusses (bullet list: "Strong adaptation to escalations"), minuses ("Overreacted to one de-escalation"), development tips ("Focus on cumulative trends; use comments for hints"). Assign score 0-100 (e.g., based on PNL % + decisions: score = (PNL / max_possible * 50) + (optimal_actions / 5 * 50); display as progress bar). Vary based on actions/path.

3.3 Design and UI Details

- **General Design:** Based on the attached Polymarket images: Dark background (#001133), white text, rounded cards, navigation (Search, Categories like Politics), market cards with percentage bars (green Yes, red No).
- **Simulation UI:** Like Polymarket, but simulated: Order book (like the first attached image), Trade buttons, timeline (slider 0-5 events), news notifications (modal popup with breakingnews.png header).
- **Interactivity:** Clickable buttons for positions, real-time price updates (use JS for calculations).

---

4. Technical Architecture for the Prototype

- **Frontend:** React.js for interactivity (order book, buttons, popups). Show the attached order book statically at start, update dynamically.
- **Backend:** Simple JS logic (no server needed in prototype – use local state for events and calculations).
- **Data:** Static: Hardcode news, price changes, comments, and order book updates from "Market and News Content Spec v1.0ent.pdf" (e.g., array of objects).
- For bolt.new: Create a clickable demo where the user can go through the entire flow. If no-code, use Bubble.io-style logic.

---

5. Roadmap (Brief After Prototype)

- Test prototype: Gather feedback.
- Expand: Add AI news in future versions.

---

6. Conclusion

This specification explains everything in detail for bolt.new to build the prototype. Submit this document to bolt.new – it should generate an interactive demo where the user can try the simulation. If you need changes, let me know!