Project Proposal: Deal or No Deal - High-Stakes Negotiation with Banker

Proposed Group Members

• Joshika Alaparthi
• Mahendra Krishna Koneru

Project Summary

The project reimagines the television classic 'Deal or No Deal' as an interactive web experience. Players select a primary briefcase, progressively open remaining cases, and negotiate with an AI-powered Banker. The Banker analyzes session data, volatility indicators, and market events to craft persuasive offers that challenge the player's risk tolerance. Responsive layouts, CSS-driven animations, and PHP-backed state management create a dynamic, high-stakes atmosphere that aligns with the objectives of the web development course.

Proposed Wireframe

Homepage details (intro): The landing screen highlights the high-stakes theme with animated briefcases, explains the game flow, and presents a prominent CTA to start a session.
Layouts: Five unique screens — Landing & Briefcase Selection, Briefcase Reveal Board, Offer Negotiation Room, Volatile Event Dashboard, and Results & Recap.
Each Page Details: The landing page introduces the rules and initializes PHP sessions. The Reveal Board lists unopened briefcases with hover transformations and color-coded tiers. The Negotiation Room displays Banker dialogue, offer history, and timed acceptance windows. The Volatile Event Dashboard surfaces market modifiers, reshuffles, or value boosts. The Results screen summarizes winnings, unused offers, and achievements while encouraging replay.
Functionality: Sessions track player selections, remaining values, offer history, and volatility events while gating navigation based on round state and Banker responses.
Transition implementation details: PHP redirects paired with CSS fade and slide transitions maintain continuity as players move between reveal, negotiation, and recap screens.
Transformations implementation details: Briefcase hover states, Banker panel flips, and value card reveals rely on CSS transform, scale, and skew animations to reinforce interactivity.
Animations implementation details: Timed keyframe animations emphasize Banker offers, countdown expirations, and volatile events triggered by PHP session updates.
User Experience: The flow promotes tension through progressive disclosure, contextual tips, and persistent access to Banker history, allowing informed decision-making without overwhelming the player.

Dynamic Mechanics & Features

Volatile Market Events: PHP-driven random events inject multipliers, penalties, or

value swaps mid-round to keep the risk profile fluid.
Banker Strategy Engine: Multi-variable calculations consider expected value, volatility tiers, player behavior patterns, and bluff probabilities to produce offers that sometimes expire unless accepted promptly.
Progressive Value Revelation: Server logic validates each reveal, updates remaining distributions, and animates discoveries with CSS sequences synced to PHP refresh logic.
Dynamic Round Structure: Non-linear round sizes allow sudden mid-game reassignment of case values while maintaining audit logs in the session.

Test Cases

✓ Conduct usability testing with focus groups to gather feedback on navigation, negotiation comprehension, and perceived pressure.
✓ Test responsiveness on phones, tablets, and desktops, ensuring animations and briefcase interactions remain smooth.
✓ Validate Banker algorithm edge cases, including extreme volatility values, rapid accept/reject decisions, and nearly exhausted briefcase sets.
✓ Confirm session integrity across refreshes, back navigation, and multi-tab usage, ensuring offer history and market events persist.

Approach to Refine Improvements

Approach to Refine Improvements: Incorporate user feedback to optimize the UI and streamline the experience.
Iterate on wireframes and prototypes based on usability results to ensure intuitive navigation and clear information presentation.
Evaluate analytics on event frequency, Banker satisfaction rates, and player retention to tune volatility and pacing.
Document revisions and update test cases to cover new behaviors introduced through iterative design.

Proposed APIs

Primary functionality relies on PHP session management and hardcoded value sets validated server-side.
Optional integration with a financial sentiment API (e.g., Finnhub) can modulate volatility events based on real-world indicators.
Email or SMS services (e.g., Twilio SendGrid) may deliver post-game summaries or tournament invitations.

Implementation Overview

Backend Implementation: PHP scripts manage sessions, case assignments, Banker offers, volatility triggers, and validation.
Frontend Implementation: HTML templates with modular CSS handle responsive layouts, transformations, transitions, and keyframe animations.
Data Management: Hardcoded briefcase values and volatility presets load at session start while offer history persists in the session scope.
Tooling: Utilize Git for version control, automated PHP unit tests for Banker logic, and Lighthouse audits for performance and accessibility.

Statement of Commitment

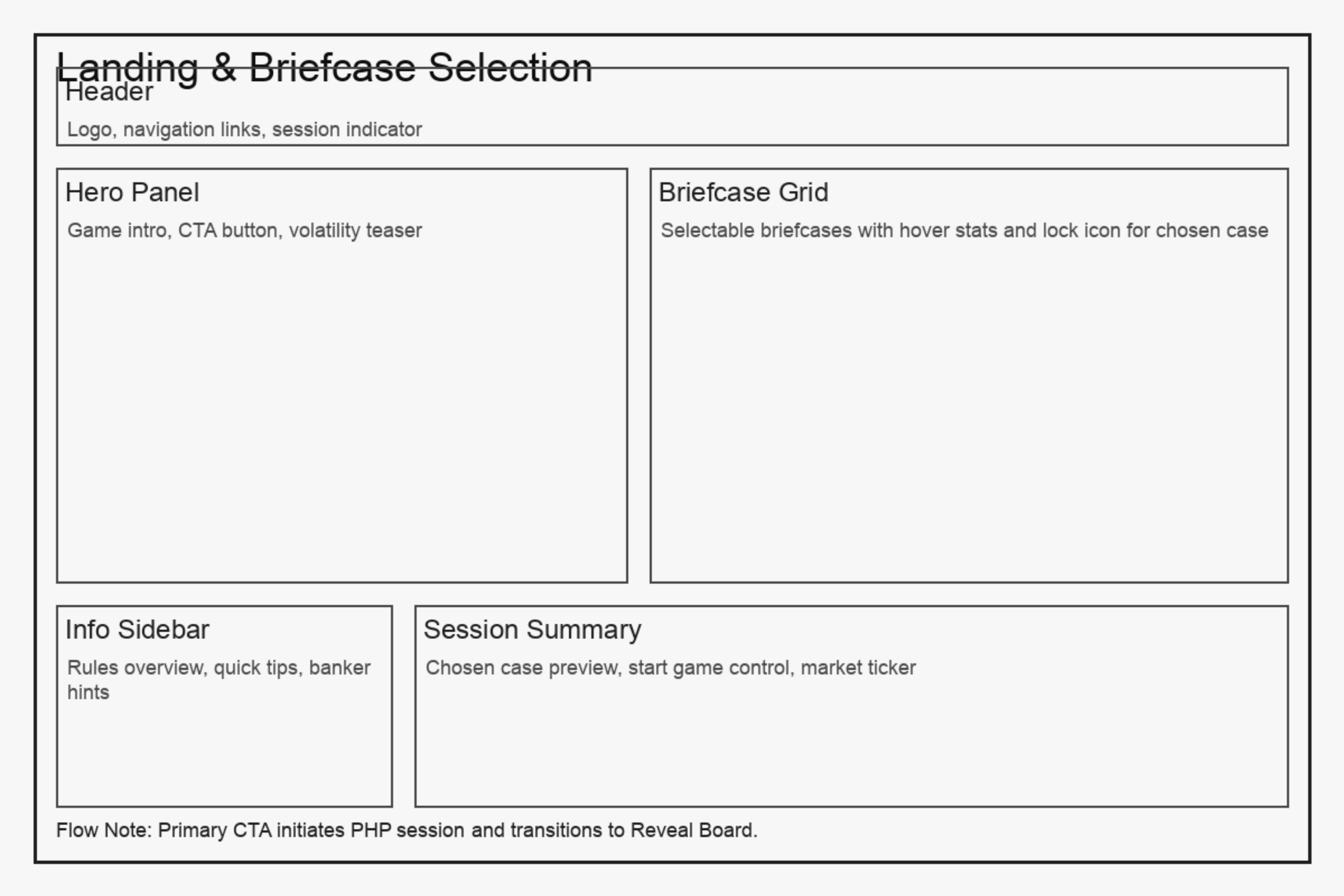
We, the undersigned members of the group, acknowledge that we understand the conditions outlined in this proposal. We are committed to actively participating in the implementation of the group project from start to finish. We understand that individual grades may vary based on our contributions and performance, and we agree to present an important phase of the work collectively.

Signed proposal:
Name: FN_____________ _____________ LN _____________ _____________ [   ]
Name: FN_____________ _____________ LN _____________ _____________ [   ]
Name: FN_____________ _____________ LN _____________ _____________ [   ]
Date: _______________________

Wireframe Appendix

The following pages contain the visual wireframes for each primary screen.

Wireframe: Landing & Briefcase Selection



Landing & Briefcase Selection

Header

Logo, navigation links, session indicator

Hero Panel

Game intro, CTA button, volatility teaser

Briefcase Grid

Selectable briefcases with hover stats and lock icon for chosen case

Info Sidebar

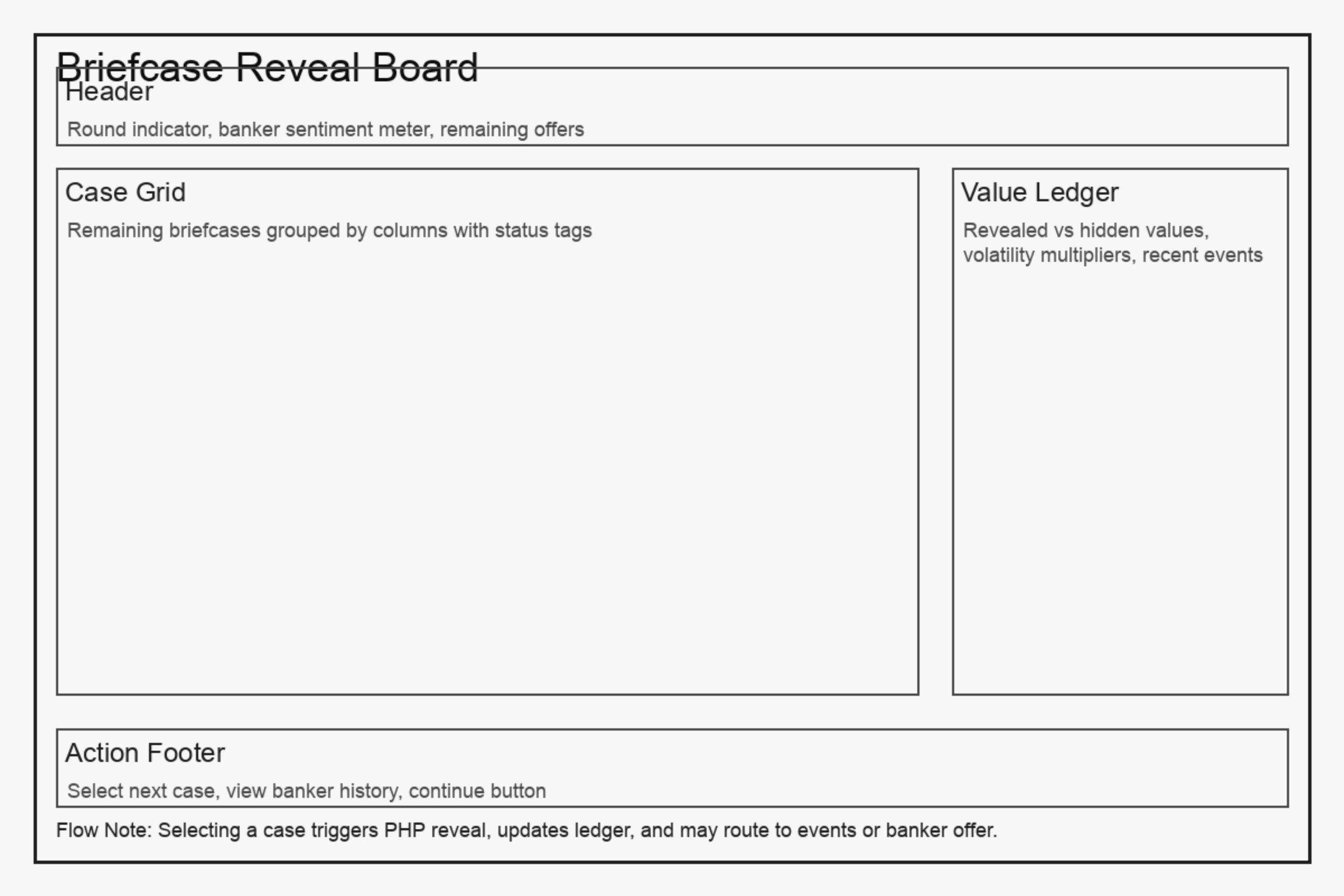
Rules overview, quick tips, banker hints

Session Summary

Chosen case preview, start game control, market ticker

Flow Note: Primary CTA initiates PHP session and transitions to Reveal Board.

Wireframe: Briefcase Reveal Board



Briefcase Reveal Board

Header

Round indicator, banker sentiment meter, remaining offers

Case Grid

Remaining briefcases grouped by columns with status tags

Value Ledger

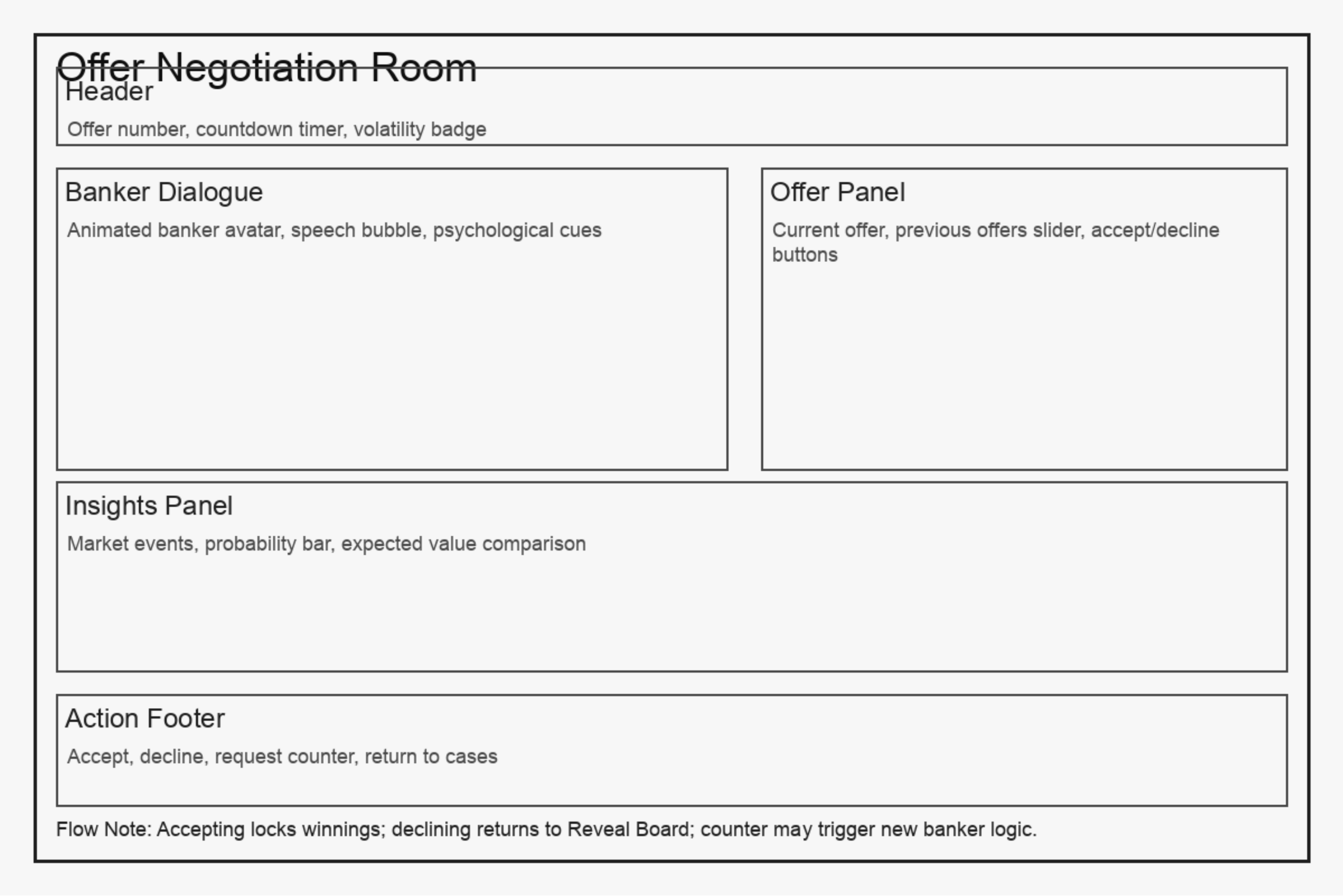
Revealed vs hidden values, volatility multipliers, recent events

Action Footer

Select next case, view banker history, continue button

Flow Note: Selecting a case triggers PHP reveal, updates ledger, and may route to events or banker offer.

Wireframe: Offer Negotiation Room



Offer Negotiation Room

Header

Offer number, countdown timer, volatility badge

Banker Dialogue

Animated banker avatar, speech bubble, psychological cues

Offer Panel

Current offer, previous offers slider, accept/decline buttons

Insights Panel

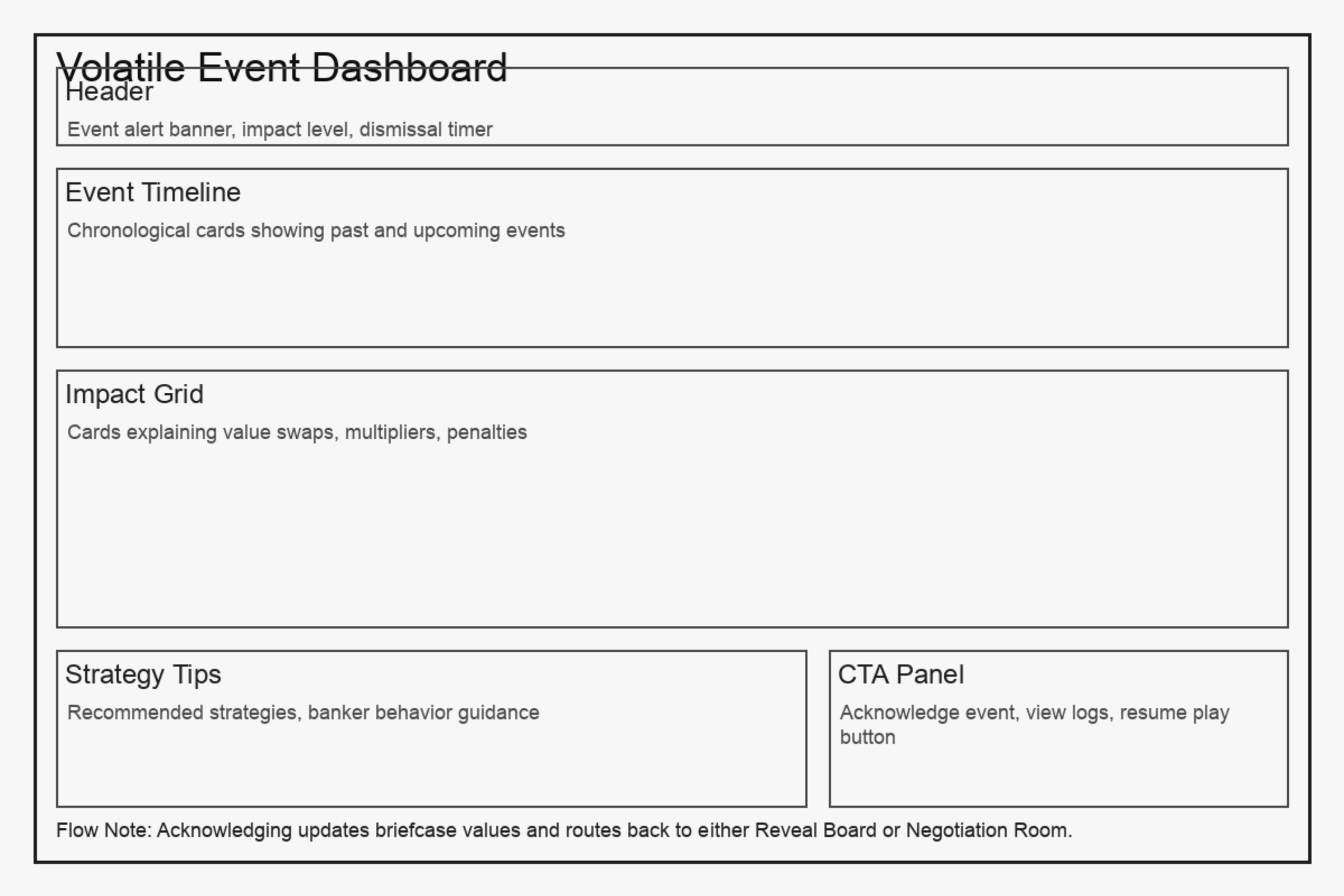
Market events, probability bar, expected value comparison

Action Footer

Accept, decline, request counter, return to cases

Flow Note: Accepting locks winnings; declining returns to Reveal Board; counter may trigger new banker logic.

Wireframe: Volatile Event Dashboard



Volatile Event Dashboard

Header

Event alert banner, impact level, dismissal timer

Event Timeline

Chronological cards showing past and upcoming events

Impact Grid

Cards explaining value swaps, multipliers, penalties

Strategy Tips

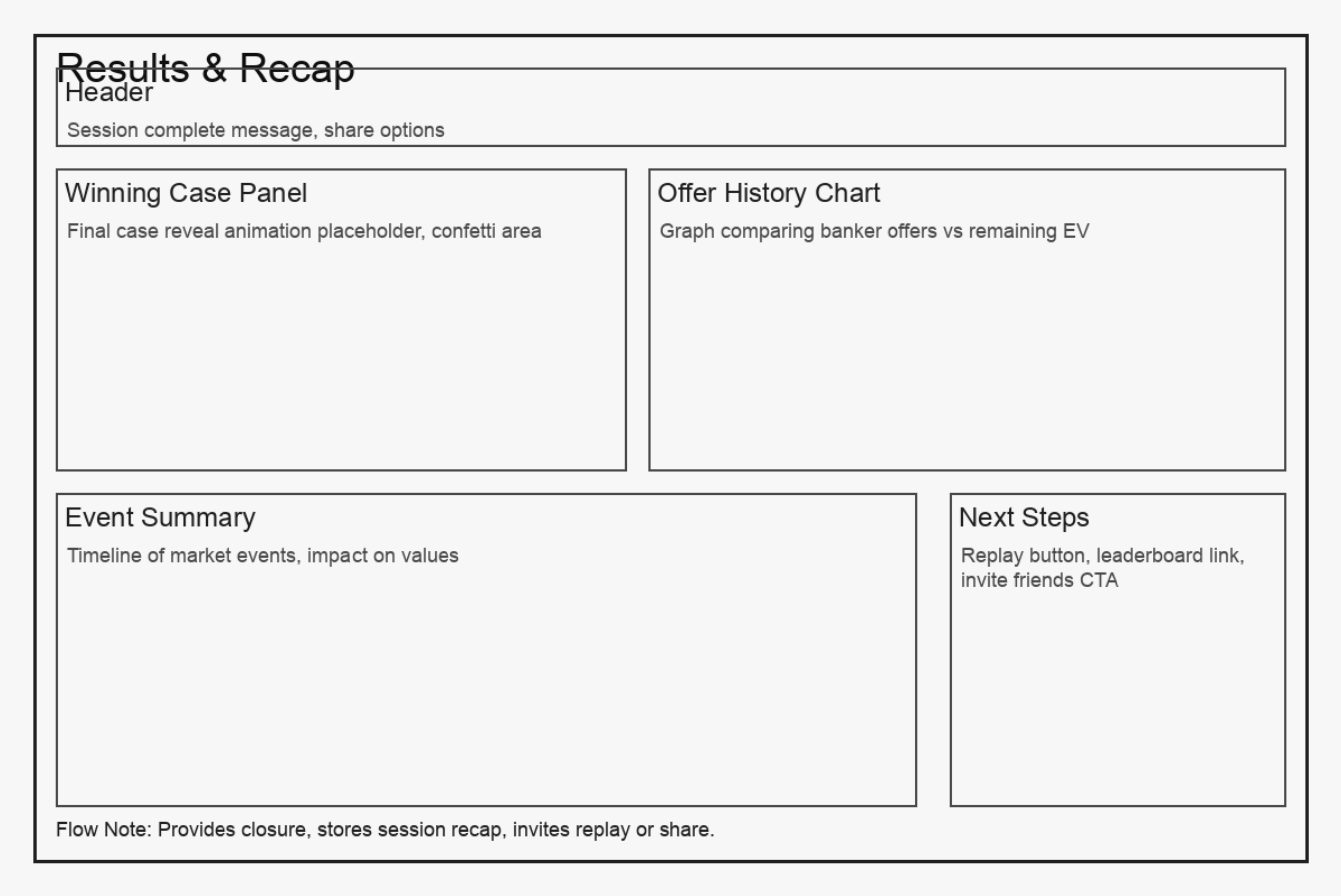
Recommended strategies, banker behavior guidance

CTA Panel

Acknowledge event, view logs, resume play button

Flow Note: Acknowledging updates briefcase values and routes back to either Reveal Board or Negotiation Room.

Wireframe: Results & Recap



Results & Recap

Header

Session complete message, share options

Winning Case Panel

Final case reveal animation placeholder, confetti area

Offer History Chart

Graph comparing banker offers vs remaining EV

Event Summary

Timeline of market events, impact on values

Next Steps

Replay button, leaderboard link, invite friends CTA

Flow Note: Provides closure, stores session recap, invites replay or share.