Minutes of Meeting (MOM) — Tic-Tac-Toe Game Project

Meeting date: 15 November 2025
Meeting start: 18:00 IST (UTC+05:30)
Meeting end: 19:10 IST (UTC+05:30)
Location / medium: Video call / Project workspace
Prepared by: ChatGPT (for use as MOM & BRD input)

Attendees

- Product Owner (PO) — Ashok Mehra
- Technical Lead (TL) — Priya Sharma
- UX Designer — Rohan Gupta
- QA Lead — Megha Rao
- Backend Engineer — Karan Singh
- Frontend Engineer — Neha Patel
- Game Logic Developer — Arjun Verma
- Stakeholder / Business Sponsor — Ms. Shalini Kapoor

Purpose / Objective (Project title & objective)

Project Title: Tic-Tac-Toe — Cross-Platform Casual Game (TICTACTOE-1)
Objective: Deliver a clean, accessible, bug-free Tic-Tac-Toe game supporting single-player (AI), local two-player (same device), and online two-player modes; provide an extendable codebase and clear acceptance criteria so an LLM can generate a BRD and epics/stories from this MOM.

Agenda

1. Clarify business goals and target platforms.
2. Define scope (in/out).
3. Capture business & functional requirements.
4. Define acceptance criteria and E2E workflow.
5. Identify stakeholders, assumptions, constraints, risks, dependencies.
6. Agree action items, owners, and deadlines.

Executive Summary / Key Decisions

- Target platforms: Web (React/HTML), Mobile (optional PWA / future native). Initial release: **Web PWA**. Decision made at **18:20 IST**.
-

Game modes for MVP: **Single-player (AI with two difficulty levels)** and **Local two-player (same device)**. Online multiplayer deferred to MVP+ (phase 2) but basic matchmaking API stub included. Decision at **18:32 IST**.

- UI: Minimal, accessible design; keyboard + mouse + touch support. Decision at **18:40 IST**.

- Storage: Use localStorage for player preferences & scores in MVP; server persistence (user accounts, global leaderboard) is out of scope for MVP. Decision at **18:45 IST**.

- Delivery target for playable prototype (MVP): **22 November 2025, 17:00 IST** (Action item).

Business Requirements (high-level)

1. Provide a simple, engaging tic-tac-toe experience accessible from modern browsers and mobile devices via PWA.

2. Support quick sessions (<= 2 minutes per match).

3. Offer single-player mode with two AI difficulties: *Easy* (random/combinational) and *Hard* (minimax or equivalent optimal play).

4. Allow two humans to play on the same device (alternating turns).

5. Track per-device win/loss/draw stats and display after each match.

6. Provide clear rules and visual indicators (current player, winning line, draw).

7. Provide accessibility features (screen-reader friendly, high-contrast mode, keyboard

navigation).

8. Instruments for basic analytics/events (matches started/finished, results) for future improvement (privacy-compliant).

Functional Requirements (detailed)

Core gameplay

- FR-01: Board: 3x3 grid with cells selectable.
- FR-02: Players: Player X and Player O. Option for player to choose symbol.
- FR-03: Turn management: enforce alternating turns; prevent selecting already-occupied cells.
- FR-04: Win detection: detect horizontal/vertical/diagonal 3-in-a-row and highlight winning cells.
- FR-05: Draw detection: board full with no winner triggers draw state.
- FR-06: Undo move: optional in MVP? — **Deferred** (MVP excludes Undo).
- FR-07: AI Modes:
  - Easy: choose a random available cell (with optional heuristics like block immediate win).
  - Hard: use minimax (or equivalent) guaranteeing optimal play; must be efficient (< 50 ms

on typical mobile).

- FR-08: Local two-player: same-device alternate taps/clicks.
- FR-09: UI states: Idle, In-Game, Game-Over (Win/Draw), Settings, Stats.
- FR-10: Settings: choose mode, AI difficulty, player symbol, toggle sounds, toggle accessibility features.
- FR-11: Stats: per-device stats (Wins/Losses/Draws/Total Matches) shown in a Stats panel.
- FR-12: New Game / Reset: start new match or reset stats (with confirmation).
- FR-13: PWA: installable, offline playable for local modes.
- FR-14: Analytics events: emit lightweight events (anonymous) for basic metrics (opt-in).

Optional / Phase 2 (MVP+)

- Online multiplayer with matchmaking and basic user accounts.
- Leaderboard (global) and persistent server stats.
- Undo/replay / move history / save & resume.
- Custom themes / skins.

Non-Functional Requirements

- NFR-01: Performance — UI frame updates < 16ms; AI move computation < 50ms on mid-range mobile.
- NFR-02: Accessibility — WCAG AA conformance for core flows.
- NFR-03: Responsiveness — works on 320px to 1920px widths; touch & keyboard friendly.
- NFR-04: Security & Privacy — no PII in analytics; opt-in for telemetry; local storage only for MVP.
- NFR-05: Maintainability — modular code, 80% unit-test coverage for game logic.
- NFR-06: Cross-browser support — latest versions of Chrome, Firefox, Safari, Edge (desktop & mobile).
- NFR-07: Robustness — graceful handling of invalid inputs and offline state for local play.

Scope — Included / Excluded (explicit)

Included (MVP):

- Web PWA playable in browser and installable.

- Single-player (Easy/Hard AI).
- Local two-player (same device).
- Local stats, settings, accessibility options.
- Basic analytics (anonymous, opt-in).
- Win/draw detection, UI flow, and game reset.

Excluded (MVP):

- Online multiplayer (full matchmaking, real-time networking).
- Persistent server-side accounts and leaderboards.
- Undo move, move history, or replay feature (deferred).
- Complex AI tuning or learning AI.
- Monetization (ads, in-app purchases) — out of scope.

Acceptance Criteria / Expected Outcomes

For **MVP release**:

1. Player can start a game and complete a match (win/draw) without errors.

2. AI Hard prevents a human from winning if played optimally (i.e., no forced-losing states unless user makes mistakes).

3. Win detection reliably highlights winning line and displays correct result within 200ms.

4. Local two-player mode allows alternating moves and enforces rules.

5. Settings persist across sessions (via localStorage).

6. PWA install prompt appears on supporting devices; offline play works for local modes.

7. Accessibility: can play using keyboard only; screen reader reads cell info and game state.

8. Unit tests for all game logic paths (win/draw detection, move validation, AI move) exist and pass.

9. Cross-browser smoke tests pass on the three major browsers.

End-to-End Workflow (detailed step-by-step)

1. **Launch / Entry**

   - User opens URL or PWA tile (time: ~0s).

- App loads, reads saved settings & stats from localStorage, shows Home screen.

2. **Configure / Mode selection**

   - User chooses mode: Single Player / Local Two Player.
   - If Single Player, user selects difficulty (Easy/Hard) and selects symbol (X/O).

3. **Start Game**

   - User taps/clicks Start. Board shows empty 3x3. Current player indicator shows whose turn.

4. **Gameplay Loop**

   - On user action (tap/click/keyboard), validate cell is empty; update board; check for win/draw.
   - If game not over, switch turn. If AI mode and AI turn, compute AI move and apply.
   - After each move, emit analytics event (if opt-in) and update in-memory match state.

5. **End of Game**

   -

If win: highlight winning line, display "Player X wins" or "AI wins". Update local stats.

- If draw: display "Draw"; update local stats.
- Offer actions: New Game, Back to Home, Share Result (optional), Reset Stats.

6. **Post-game**

   - Stats updated in persistent localStorage. User can view Stats screen for aggregated results.

7. **Settings & Accessibility**

   - At any time user can open Settings to toggle accessibility, sound, change AI difficulty. Changes persist.

8. **Offline behavior**

   - If offline, Single Player & Local play are fully functional; analytics events queue locally until online, but user must opt-in.

User Stories (enough detail to create stories + acceptance criteria)

(IDs are for traceability when converting to BRD / Jira)

Epic: Core Gameplay

- US-101 — As a player, I want to start a new single-player game so I can play against AI.
  - AC: Board loads, AI difficulty selectable, game starts.
- US-102 — As a player, I want to play local two-player so I can play with a friend on the same device.
  - AC: Turns alternate; occupied cells blocked; win/draw detected.
- US-103 — As a player, I want to see who’s turn it is so I don’t make illegal moves.
  - AC: Turn indicator updates after each valid move.
- US-104 — As a player, I want the game to detect and highlight a winning line.
  - AC: On win, the 3 winning cells are highlighted and a message displays.
- US-105 — As a player, I want the game to detect draws and notify me.
  - AC: On full board with no winner, draw state presented.

Epic: AI

- US-201 — As a player, I want an Easy AI so I can have casual matches.
  - AC: AI picks random/heuristic move; behaves non-deterministically.
- US-202 — As a player, I want a Hard AI so I can be challenged.
  - AC: AI performs optimal move (minimax), cannot be forced to lose if user plays optimally.

Epic: Settings & Persistence

- US-301 — As a player, I want my settings saved so I don't reconfigure each session.
  - AC: Settings persist in localStorage and apply on next load.

Epic: Accessibility & UX

- US-401 — As a player using keyboard or screen reader, I want to navigate and play without a mouse.
  - AC: Keyboard navigation, ARIA labels, screen reader announces cell states.

Epic: QA / Testing

- US-501 — As QA, I need unit tests for game logic so regressions are avoided.
  - AC: Tests for win/draw detection, illegal moves, AI move validity, all pass.

(Each US should be split into tasks during sprint planning; acceptance criteria above are minimal and must be included in BRD-derived stories.)

Assumptions

- Target users have modern browsers and devices with basic JS support.
- No user authentication required for MVP.
- No monetization for initial release.
- Analytics only if user opts in (privacy-first).
- AI Hard can be implemented client-side without heavy CPU costs for 3x3 grid.

Constraints & Dependencies

- Constraint: MVP must be single-repo, small footprint (< 1 MB compressed assets where possible).

- Dependency: Browser storage (localStorage) availability.
- Dependency: CI pipelines for unit tests; test runner (Jest) to be decided.
- Constraint: No external server needed for MVP; if server chosen, must be optional.
- Dependency (phase 2): Backend services for accounts/leaderboard.

Risks & Mitigations

1. **Risk:** Hard AI may cause noticeable delay on low-end devices.
   - Mitigation: Use optimized minimax with alpha-beta pruning or precomputed strategy; cap compute time; fallback to Easy if timing threshold exceeded.
2. **Risk:** Accessibility not properly implemented, causing WCAG failures.
   - Mitigation: Early UX testing with screen reader and keyboard-only flows; include accessibility checklist in Definition of Done.
3. **Risk:** Browser storage not available (privacy mode).
   - Mitigation: Graceful degrade: session-only state and user notification that stats won’t persist.

4. **Risk:** Scope creep (online play).

   - Mitigation: Strictly freeze MVP scope; move online features to roadmap/phase 2.

Action Items (owner — due date & time IST)

1. **Prototype UI (clickable)** — UX (Rohan) — **Due: 18 Nov 2025, 12:00 IST**

   - Deliver: Home, Game screen, Settings, Stats wireframes; include keyboard flow and ARIA notes.

2. **Implement game logic module (win/draw detection & move validation)** — Game Logic Dev (Arjun) — **Due: 19 Nov 2025, 18:00 IST**

   - Deliver: Unit-tested module with ≥ 95% logic coverage.

3. **Implement AI Easy & Hard** — TL & Game Logic (Priya + Arjun) — **Due: 20 Nov 2025, 18:00 IST**

   - Deliver: Easy (random/heuristic) and Hard (minimax w/ pruning) with performance tests.

4. **Frontend integration (React PWA)** — Frontend (Neha) — **Due: 21 Nov 2025, 17:00 IST**

- Deliver: UI wired to game logic + settings persistence + responsive layout.

5. **QA test plan & test cases** — QA (Megha) — **Due: 21 Nov 2025, 23:59 IST**
    - Deliver: Test cases for functional & accessibility acceptance criteria.

6. **Analytics opt-in design & privacy doc** — PO (Ashok) — **Due: 20 Nov 2025, 23:59 IST**
    - Deliver: Short privacy statement & list of events tracked (opt-in).

7. **Playable MVP demo** — All — **Due: 22 Nov 2025, 17:00 IST**
    - Deliver: Deployable PWA build and demo script.

8. **BRD generation from MOM** — PO / LLM (user will run LLM) — **Due: 23 Nov 2025, 12:00 IST**
    - Deliver: Business Requirements Document for stakeholder sign-off.