Bingo Game System - Developer Brief

See attached Excel Sheet

See:

Bingoforever.world/ops

Bingoforever.world/join

Overview:
The goal is to build an interactive bingo game that manages participants, assigns cards randomly, and automatically marks numbers during gameplay. The system must be scalable, clear to operate, and easily maintained.

Key Components

1. Operator User (Admin)

- The Operator controls the game from a dedicated webpage (/ops).
- The Operator has access to four key actions:
    - **Open Game** — Allows players to join.
    - **Close Entries** — Prevents additional players from joining.
    - **Start Game** — Distributes bingo cards to players and begins automated number calling.
    - **Stop and Reset** — Ends the game, clears all data, and prepares for the next session.

2. Player User (Participants)

- Players join the game via a dedicated webpage (/play).
- Before the Operator opens the game, players see: "*Game not yet open.*"
- After the game is opened, players see an **Enter Game** button.
- After clicking 'Enter Game', players see: "*You're in the game. Please wait for the game to start.*"

Database Requirements

The database must manage three key data sets:

1. Participants

- Each player who clicks 'Enter Game' should have their unique identifier stored in a participants list.
- This list is dynamic and must track the number of participants for the Operator to monitor.

2. Card Data

- **CARD DATA 1** (and additional CARD DATA sets) will be provided as JSON files.
- Each file will contain:
    - **Row 0** — The **20 CALL NUMBERS** that will be announced during the game.
    - **Rows 1+** — Each row contains **16 numbers** (forming a 4x4 bingo card).
- The system must assign bingo cards as follows:
    - If **60 players** enter the game, the system must distribute **Rows 1-60** to participants.
    - Assignment must be **randomized** (players should not receive cards based on their entry order).

3. Assigned Cards

- When the Operator presses **Start Game**, the system must:
    - Randomly distribute cards from the Card Data set to all participants.
    - Ensure **Row 1** (the winning card) is assigned randomly.

Game Logic

Start Game Sequence

- When the Operator presses **Start Game**:
    - Each participant's /play page should refresh to display their assigned card as a **4x4 grid**.
    - Numbers should be automatically 'marked off' as they are called.

Call Numbers Sequence

- After pressing **Start Game**, a countdown begins:
    - **26-second delay** before the first number appears.
    - Each subsequent number appears **every 6 seconds**.
- Each participant's card will automatically highlight matching numbers.

Win Condition

- When the final number is called:
    - The participant assigned **Row 1** must see: "🎉 *WINNER!* 🎉"
    - All other participants will see: "*Not you this time.*"

Reset Game

- When the Operator clicks **Stop and Reset**:
    - All participants are removed from the database.
    - The system prepares for a new game.
    - The system should check if a new CARD DATA file is available. If not, it reuses CARD DATA 1.

Key Notes for the Developer

☑ The system must function independently of specific platforms like Firebase or Cloudinary. The developer may choose the most suitable storage/database solution.
☑ Participant data must persist across browser refreshes.
☑ Assigned cards should refresh automatically when the game starts.
☑ Security and data integrity should be prioritized — only the Operator should control game flow.

This streamlined brief provides clear instructions without dictating technical implementation. Let me know if you'd like to refine any sections.